NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

CALCULATION OF CHIP TEMPERATURES
USING
ELLPACK

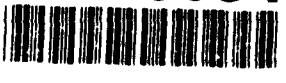
by

Vincent J. van Joolen

June, 1991

Thesis Advisor: Beny Neta

Approved for public release; distribution is unlimited



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (*If applicable*) 55	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	7b. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10. SOURCE OF FUNDING NUMBERS Program Element No. / Project No. / Task No. / Work Unit Accession Number

11. TITLE (*Include Security Classification*)
CALCULATION OF CHIP TEMPERATURES USING ELLPACK

12. PERSONAL AUTHOR(S) van Joolen, Vincentius J.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*year, month, day*)	15. PAGE COUNT
Master's Thesis	From To	June 1991	192

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (*continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUBGROUP	ELLPACK, heat transfer interfaces.

19. ABSTRACT (*continue on reverse if necessary and identify by block number*)

In this thesis we report on an experiment to solve a three dimensional elliptic problem using ELLPACK software. The physical problem arises when calculating the distribution of temperatures within an electronic package in order to assist in the design of appropriate cooling. ELLPACK software is modified here to solve three dimensional problems more general than the ones it was originally designed for.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS REPORT ☐ DTIC USERS	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Beny Neta	22b. TELEPHONE (*Include Area code*) (408) 646-2235	22c. OFFICE SYMBOL MA/Nd

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited.

Calculation of Chip Temperatures
using ELLPACK

by

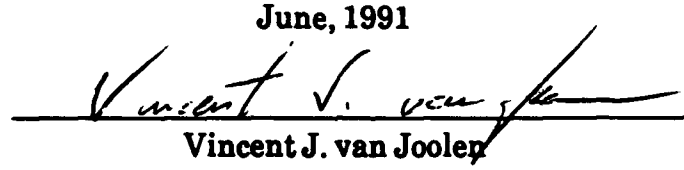
Vincent J. van Joolen
Lieutenant, United States Navy
B.S., University of California, San Diego, 1983

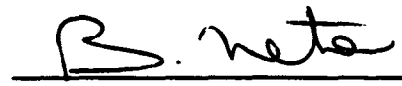
Submitted in partial fulfillment
of the requirements for the degree of

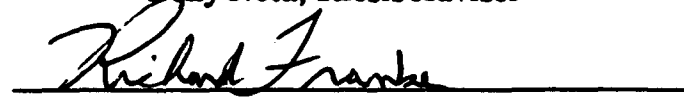
MASTER OF SCIENCE IN APPLIED MATHEMATICS

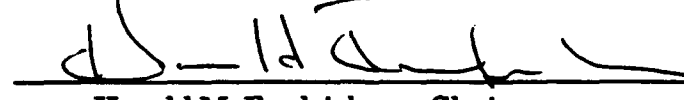
from the

NAVAL POSTGRADUATE SCHOOL
June, 1991

Author: 
Vincent J. van Joolen

Approved by: 
Beny Neta, Thesis Advisor


Richard H. Franke, Second Reader


Harold M. Fredricksen, Chairman
Mathematics

ABSTRACT

In this thesis we report on an experiment to solve a three dimensional elliptic problem using ELLPACK software. The physical problem arises when calculating the distribution of temperatures within an electronic package in order to assist in the design of appropriate cooling.

ELLPACK software is modified here to solve three dimensional problems more general than the ones it was originally designed for.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

ACKNOWLEDGMENT

To my advisor, Professor Beny Neta, I wish to express deep appreciation for his guidance in my research. To Professor Richard Franke, I wish to thank you for the very thorough first exposure you gave me of the use of numerical methods to solve partial differential equations.

To my father, Theodore, who passed away shortly before I arrived at the Naval Postgraduate School, I wish to express my sincere gratitude for instilling in me the importance of pursuing a higher education. Finally, I wish to thank my wife, Beverly, whose love, patience, and understanding made this thesis possible.

I. PRELIMINARIES

A. INTRODUCTION

The ultimate objective of this thesis is to provide thermal designers with the necessary tools and algorithms to estimate steady state component temperatures in geometries involving convective cooling of electronic equipment. The need for this capability becomes increasingly important as energy generation rates within modern electronic packages continue to rise considerably. Suitable cooling schemes must be devised to ensure that these components operate reliably.

The scope of the computations involved are appreciated when a typical configuration is considered in Figure 1.1. A liquid filled enclosure with a number of electronic packages mounted on a substrate is cooled at the top using a thermoelectric element. The energy generated within the electronic packages is transferred to the substrate and to the surrounding fluid medium by means of conduction and radiation. If the enclosure wall and bottom are well insulated, and if there is no change in phase in the fluid medium, this energy is than transported by means of natural convection to the top surface where it is removed by the thermoelectric element.

The transport mechanism responsible for the removal of the heat generated from the box is identified when considering the construction of a typical box portrayed in Figure 1.2. As previously stated, the heat generated within the chip is removed by a combination of conduction and radiation, however, the effects of radiation are negligible within the

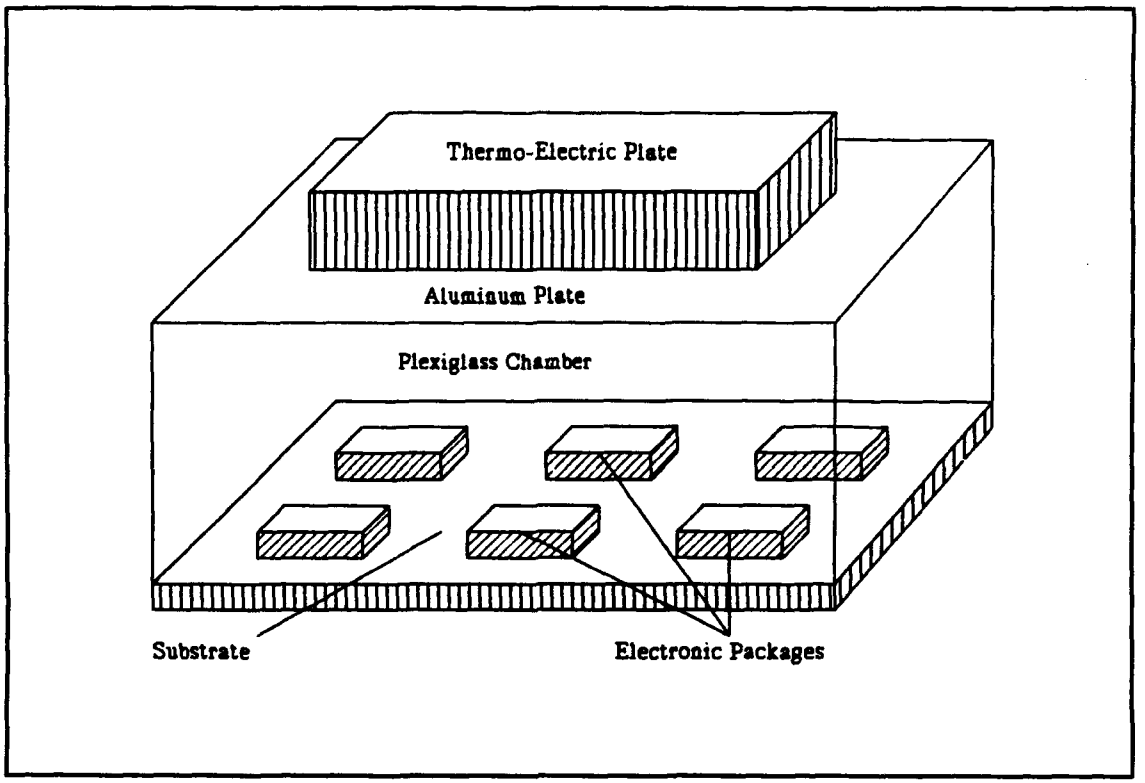


Figure 1.1: Schematic sketch of an experimental arrangement for a liquid immersion cooling of an array of electronic packages.

experimental temperature ranges of this problem. Since conduction is the primary mode of heat transfer, the internal temperature of the electronic complex is governed by elliptical partial differential equations that are affected by convective boundary conditions on the chip and substrate. Solving this problem necessitates the assignment of thermal properties, based on laboratory measurements, to the material within the electronic package. This data is then utilized to numerically solve for the temperature distribution inside the region.

The mathematical model is described in the next section of this chapter. Chapter II introduces the use of finite difference equations derived from Taylor series polynomials

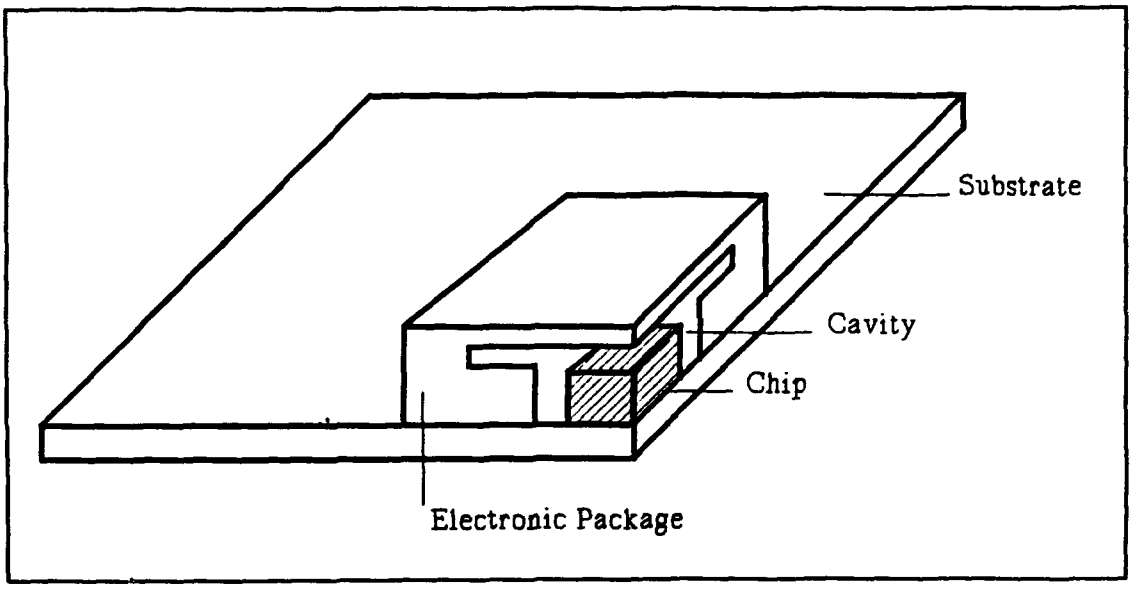


Figure 1.2: Simplified representation of an electronic package mount on a substrate.

to numerically approximate the solution to partial differential equations. Solving partial differential equations in this manner is accomplished using available ELLPACK software which is introduced in Chapter III. In Chapters IV and V numerical algorithms are derived to model situations that arise from the configuration under investigation that can not be represented using existing ELLPACK routines. In Chapter VI a program is introduced that incorporates the equations derived in Chapter IV and V into the ELLPACK system. Several experiments are then performed checking the validity of the algorithms developed. Chapter VI also provides an approximation for the temperature profile of the electronic complex described in Figure 1.2. This paper concludes in Chapter VII with suggestions for future research.

B. MATHEMATICAL MODEL

The simplified model under consideration describes the heat transfer in a complex of a single electronic package, or chip, mounted on a substrate as shown in Figure 1.3. The following material properties are defined:

- k_1: Thermal conductivity of the chip.
- k_2: Thermal conductivity of the substrate.
- h_i: Thermal convection coefficient along convective boundaries. The subscripted "i" is used to indicate that the value for h may vary along different convective boundaries.
- $\dot{q}$: Heat generated inside the chip per unit volume.

Dimensions depicted in Figure 1.3 used in this paper are:

- T_1: Thickness of the electronic chip.
- T_2: Thickness of the substrate. Typically $T_2 << T_1$.
- L_1: Length of the electronic chip.
- L_2: Width of the electronic chip.
- L: Length and width of the substrate.

To determine the temperature distribution for the complex depicted in Figure 1.4, Laplace's equation is solved for the substrate and Poisson's equation is solved for the chip. If u is the function representing temperature profile these equations are respectively:

$$\nabla^2 u = 0 \qquad |x| \le \frac{L}{2},\ |y| \le \frac{L}{2},\ 0 \le z \le T_2 \tag{1.1}$$

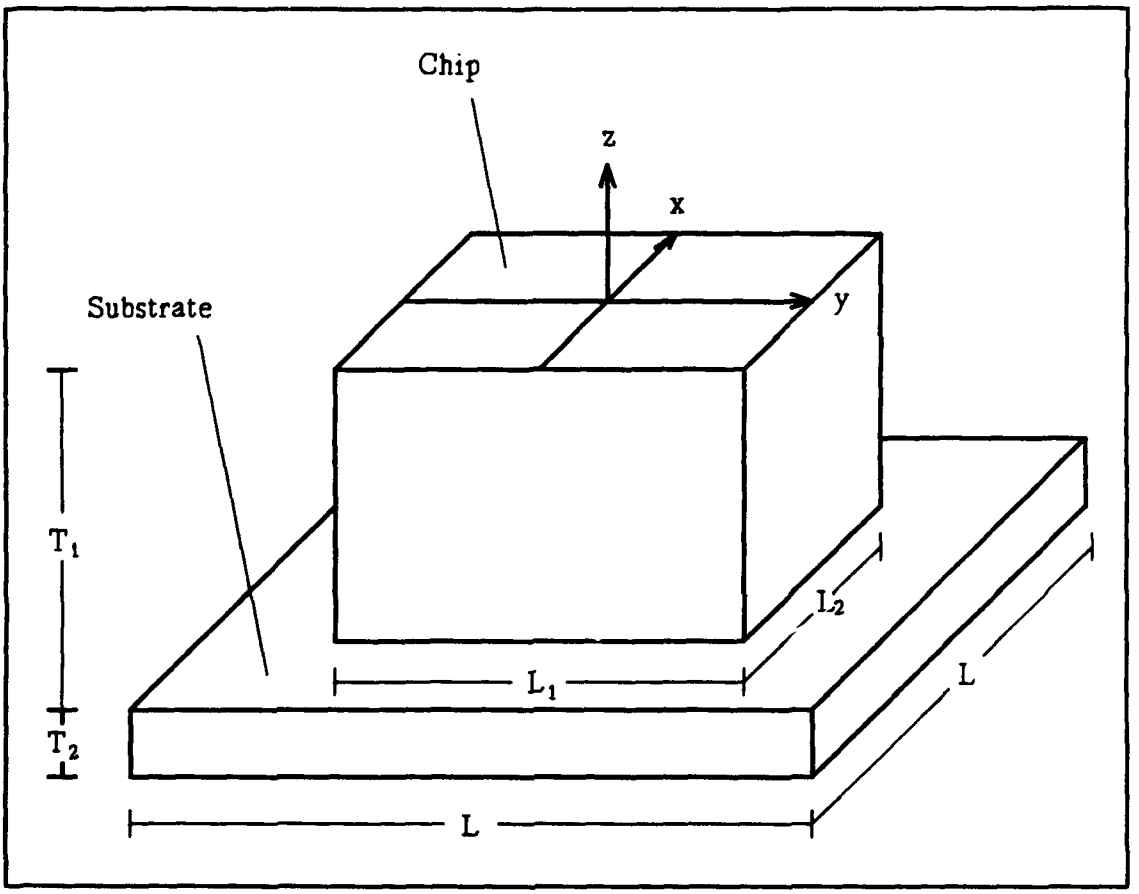


Figure 1.3: Simplified model depicting a chip on a substrate.

and

$$\nabla^2 u + \frac{\dot{q}}{k_1} = 0 \qquad |x| \le \frac{L_1}{2},\ |y| \le \frac{L_2}{2}, T_2 \le z \le T_1 + T_2 \ . \tag{1.2}$$

Details on the development of the partial differential equation model for the heat transfer mechanism are found in Haberman [Ref. 1:pp. 1-26].

Although the thermal properties of the two materials are different, the temperature and the heat flux along the material interface must be equal. The continuity requirement for heat flux is represented by

$$-k_2\frac{\partial u}{\partial z} = -k_1\frac{\partial u}{\partial z} \qquad |x| \le \frac{L_1}{2},\ |y| \le \frac{L_2}{2},\ z = T_2\ . \tag{1.3}$$

Based on the symmetry of the region we need consider only a quarter of the material domain. Figure 1.4 illustrates the region under investigation.

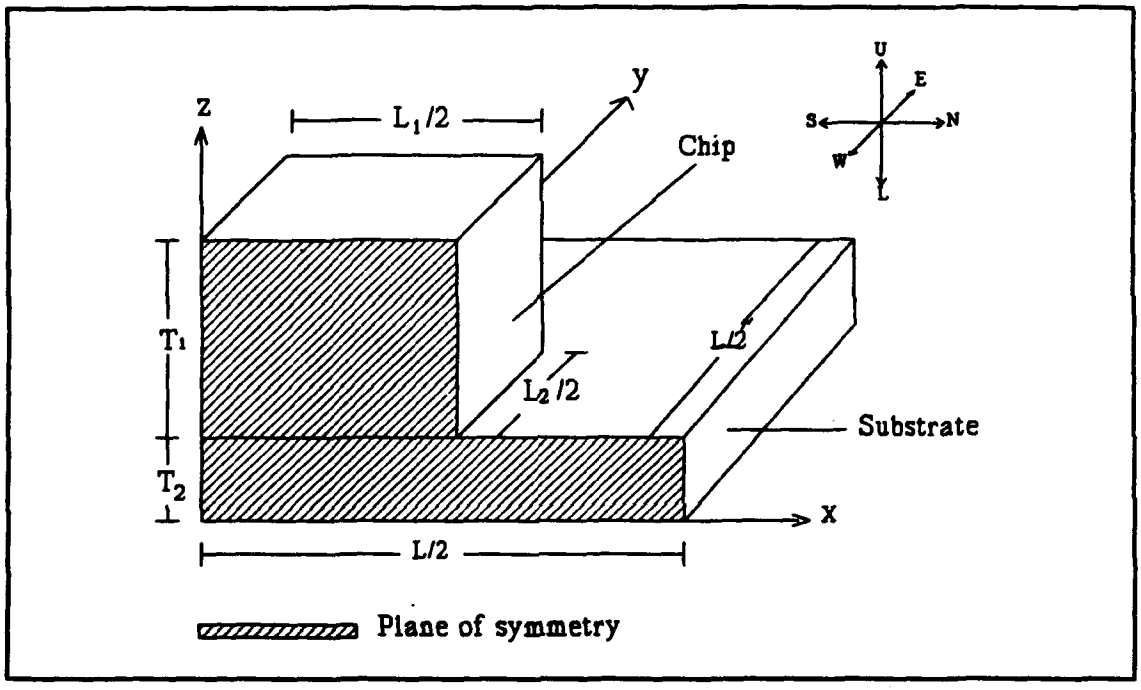


Figure 1.4: Simplified version of a chip mounted on a substrate with planes of symmetry at x = 0 and y = 0.

The boundary conditions for this region are expressed as follows:

- **Lower boundary of the substrate:**

$$-k_2\frac{\partial u}{\partial z} = h_1(u_\infty - u) \qquad 0 \le x \le \frac{L}{2},\ 0 \le y \le \frac{L}{2}, z = 0\ . \tag{1.4}$$

- Upper boundary of the substrate:

$$-k_2\frac{\partial u}{\partial z}=h_2(u-u_\infty) \qquad \frac{L_1}{2}\le x\le\frac{L}{2},\ \frac{L_2}{2}\le y\le\frac{L}{2},\ z=T_2\ . \tag{1.5}$$

- Upper boundary of the chip:

$$-k_1\frac{\partial u}{\partial z}=h_3(u-u_\infty) \qquad 0\le x\le\frac{L_1}{2},\ 0\le y\le\frac{L_2}{2},\ z=T_1+T_2\ . \tag{1.6}$$

- Eastern boundary of the substrate:

$$-k_2\frac{\partial u}{\partial x}=h_4(u-u_\infty) \qquad x=\frac{L}{2},\ 0\le y\le\frac{L}{2},\ 0\le z\le T_2\ . \tag{1.7}$$

- Eastern boundary of the chip:

$$-k_1\frac{\partial u}{\partial x}=h_3(u-u_\infty) \quad x=\frac{L_1}{2},\ 0\le y\le\frac{L_2}{2},\ T_2\le z\le T_1+T_2\ . \tag{1.8}$$

- Western plane of symmetry:

$$-k_1\frac{\partial u}{\partial x}=0 \qquad x=0,\ 0\le y\le\frac{L}{2},\ 0\le z\le T_1+T_2\ . \tag{1.9}$$

- Southern plane of symmetry:

$$-k_1 \frac{\partial u}{\partial y} = 0 \qquad 0 \le x \le \frac{L}{2},\ y = 0,\ 0 \le z \le T_1 + T_2 \ . \tag{1.10}$$

- Northern boundary of the substrate:

$$-k_2 \frac{\partial u}{\partial y} = h_6 (u - u_\infty) \qquad 0 \le x \le \frac{L}{2},\ y = \frac{L}{2},\ 0 \le z \le T_2 \ . \tag{1.11}$$

- Northern boundary of the chip:

$$-k_1 \frac{\partial u}{\partial y} = h_7 (u - u_\infty) \quad 0 \le x \le \frac{L_1}{2},\ y = \frac{L_2}{2},\ T_2 \le z \le T_1 + T_2 \ . \tag{1.12}$$

Further details pertaining to the development of boundary conditions are found in Özişik [Ref. 2:pp. 28-29].

From this discussion it is apparent that the equations governing the temperature profile for the electronic composite are too complex to solve analytically. Instead numerical methods must be employed to approximate the temperature profile inside the system.

II. NUMERICAL METHODS IN A SINGLE MATERIAL DOMAIN

A. FINITE DIFFERENCE APPROXIMATIONS ON A UNIFORM GRID

1. One Dimensional Analysis

One scheme used to approximate the solution of ordinary differential equations is to replace the derivative terms appearing in an expression by difference equations that are derived using Taylor series polynomial approximations. In Figure 2.1 a one dimensional regime is discretized by subdividing it into uniform parts of length h where x_i represents any given point on the grid. The points that lie on either side of the point x_i are indicated by

$$\begin{aligned} x_{i-1} &= x_i - h, \\ x_{i+1} &= x_i + h. \end{aligned} \tag{2.1}$$

A Taylor series expansion to three terms yields

$$u(x_{i-1}) = u(x_i) - h\frac{du}{dx}\Big|_{x_i} + \frac{h^2}{2}\frac{d^2u}{dx^2}\Big|_{x_i} - \frac{h^3}{3!}\frac{d^3u}{dx^3}\Big|_{x_i} + O(h^4), \tag{2.2a}$$

$$u(x_{i+1}) = u(x_i) + h\frac{du}{dx}\Big|_{x_i} + \frac{h^2}{2}\frac{d^2u}{dx^2}\Big|_{x_i} + \frac{h^3}{3!}\frac{d^3u}{dx^3}\Big|_{x_i} + O(h^4), \tag{2.2b}$$

where u(x) is a function that describes the temperature profile of the region. The error term in Equation 2.2 is of order h^4 and is denoted as

$$O(h^4) \propto Kh^4 \,, \tag{2.3}$$

where K is some constant. From Equation 2.3 it is apparent that the approximation is improved if the grid interval is reduced. The approximation can be further improved by using a Taylor series expansion with a greater number of terms, but for the purpose of this analysis three terms are all that are needed.

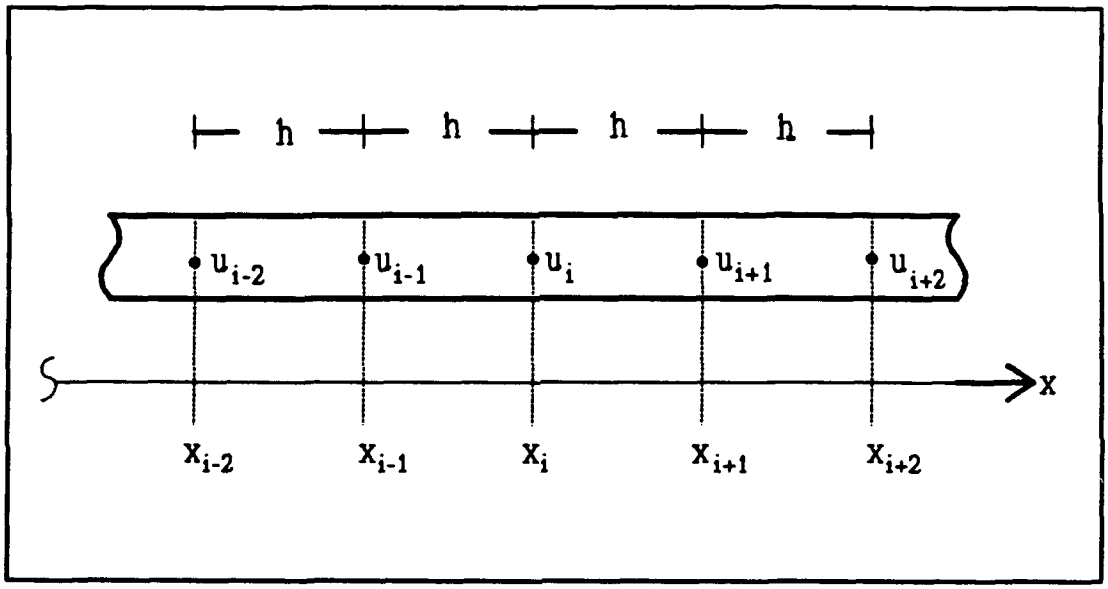


Figure 2.1: Uniform grid super-imposed on a one dimensional regime.

Using Equation 2.2 it is shown that

$$\left.\frac{d^2u}{dx^2}\right|_{x_i} = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + O(h^2). \tag{2.4}$$

Equation 2.4 is called the central difference approximation of the second derivative and is used to approximate heat transport behavior modeled by Laplace's equation (Equation

1.1) or Poisson's equation (Equation 1.2). The difference approximation for Poisson's equation in one dimension on a uniform grid is

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \frac{\dot{q}}{k} = 0. \tag{2.5}$$

This is rewritten as

$$u_{i+1} - 2u_i + u_{i-1} = -\frac{\dot{q}h^2}{k}. \tag{2.6}$$

Equation 2.6 is used to express a linear function at each node on the grid. The resulting linear system discretizes the partial differential equation and has a solution which approximates the temperature at each node. The error in the approximation that is a consequence of this discretization is of order h^2.

2. Two Dimensional Analysis

Difference approximations are also used to discretize partial differential equations that are defined on two dimensional regions. In Figure 2.2 a two dimensional region is subdivided into uniform squares with sides of length h. The function u(x,y) is the temperature function for the region. The following expansions are generated using Taylor series analysis at the grid points adjacent to the central point (x_i,y_j)

$$u_{i-1,j} = u_{i,j} - h\frac{\partial u}{\partial x}\Big|_{i,j} + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}\Big|_{i,j} - \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}\Big|_{i,j} + O(h^4), \tag{2.7a}$$

$$u_{i+1,j} = u_{i,j} + h\frac{\partial u}{\partial x}\Big|_{i,j} + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}\Big|_{i,j} + \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}\Big|_{i,j} + O(h^4), \tag{2.7b}$$

$$u_{i,j-1} = u_{i,j} - h\frac{\partial u}{\partial y}\Big|_{i,j} + \frac{h^2}{2}\frac{\partial^2 u}{\partial y^2}\Big|_{i,j} - \frac{h^3}{3!}\frac{\partial^3 u}{\partial y^3}\Big|_{i,j} + O(h^4), \tag{2.7c}$$

$$u_{i,j+1} = u_{i,j} + h\frac{\partial u}{\partial y}\Big|_{i,j} + \frac{h^2}{2}\frac{\partial^2 u}{\partial y^2}\Big|_{i,j} + \frac{h^3}{3!}\frac{\partial^3 u}{\partial y^3}\Big|_{i,j} + O(h^4), \tag{2.7d}$$

where

$$u_{i,j} = u(x_i, y_j). \tag{2.8}$$

The difference approximations for the second partial derivatives of the function u(x,y) are derived from Equation 2.7:

$$\frac{\partial^2 u}{\partial x^2}\Big|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2), \tag{2.9a}$$

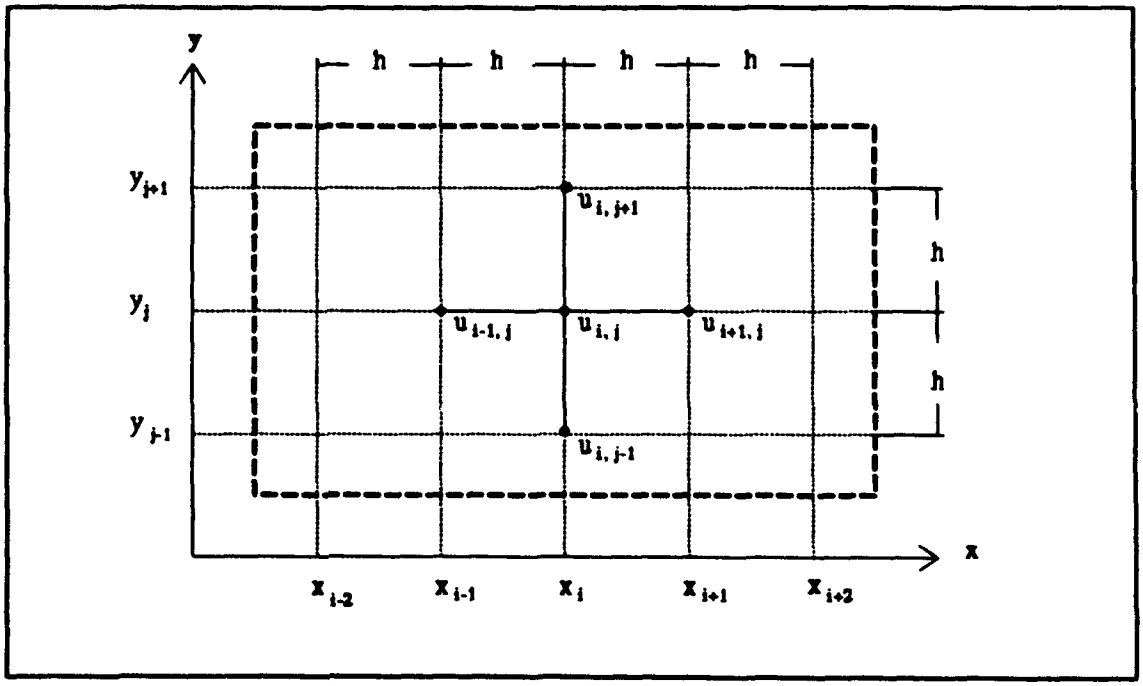


Figure 2.2: A uniform grid superimposed onto a two dimensional region.

$$\left.\frac{\partial^2 u}{\partial y^2}\right|_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + O(h^2). \tag{2.9b}$$

Substituting Equation 2.9 into Poisson's equation yields

$$u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1} = -\frac{\dot{q}h^2}{k}, \tag{2.10}$$

which is known as the "Five Point Star" equation. A system of equations is then generated by applying Equation 2.9 to each grid node. The solution of the system approximates the temperature at each node with a discretization error of order h^2.

3. Three Dimensional Analysis

The equations for three dimensions are derived by employing the same methods as those previously discussed. Figure 2.3 depicts a generalized nodal network in three dimensions where a uniform grid is used. The expression u(x,y,z) is the temperature function for the region. Using Taylor series expansions, difference approximation of the heat equation for the nodal network is

$$u_{i-1,j,k} + u_{i,j-1,k} + u_{i,j,k-1} - 6u_{i,j} + u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j,k+1} = -\frac{\dot{q}h^2}{k}. \tag{2.11}$$

Equation 2.11 is known as the "Seven Point Star" equation.

The notation used in Equation 2.11 is further abbreviated by defining the following

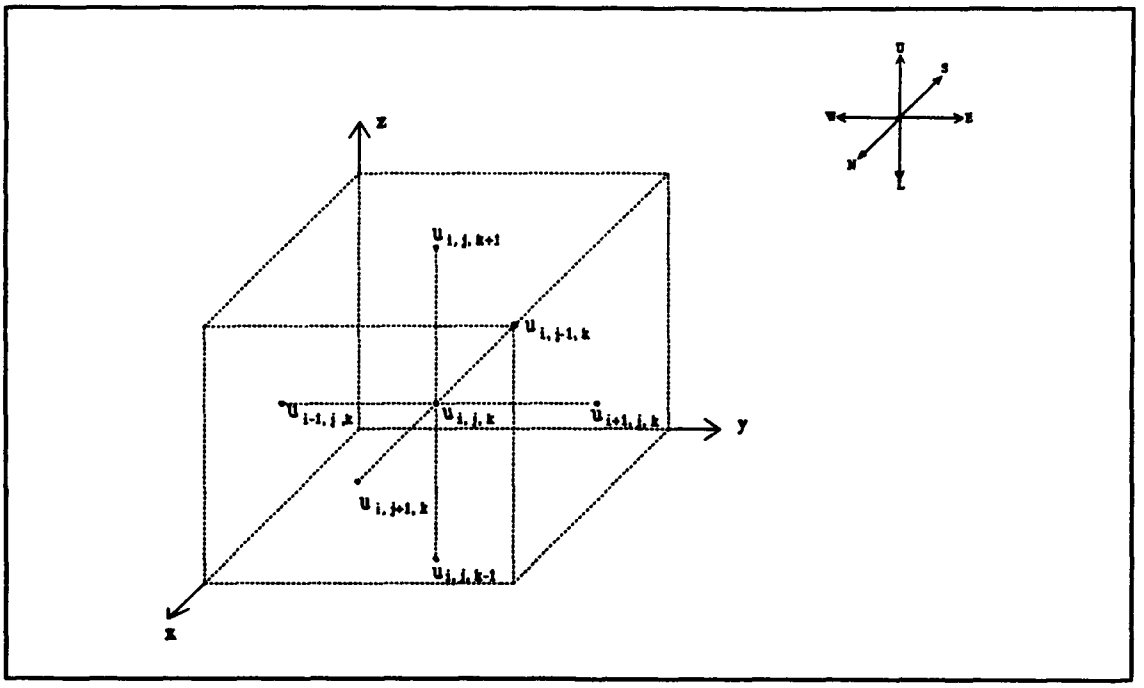


Figure 2.3: Typical nodal network on a uniform grid in a three dimensional region.

$$
\begin{aligned}
u_C &= u_{i,j,k} && \textit{(central point of nodal network)},\\
u_W &= u_{i-1,j,k} && \textit{(western point of nodal network)},\\
u_S &= u_{i,j-1,k} && \textit{(southern point of nodal network)},\\
u_L &= u_{i,j,k-1} && \textit{(lower point of nodal network)},\\
u_E &= u_{i+1,j,k} && \textit{(eastern point of nodal network)},\\
u_N &= u_{i,j+1,k} && \textit{(northern point of nodal network)},\\
u_U &= u_{i,j,k+1} && \textit{(upper point of nodal network)}.
\end{aligned}
\tag{2.12}
$$

This notation will be used throughout the paper. Using the shortened notation, Equation 2.11 is rewritten as

$$u_W + u_S + u_L - 6u_C + u_E + u_N + u_U = -\frac{\dot{q}h^2}{k}. \qquad (2.13)$$

In discussions of a one dimensional region, the upper and lower terms (or z-axis terms) and the northern and southern terms (or y-axis terms) will be omitted. For two dimensional regions the z-axis terms will be dropped.

4. Numerical Approximations of Boundary Conditions

Special consideration must be given to the nodes that lie on the boundary of a region when using a linear system to discretize a partial differential equation. Three major classes of boundary conditions are defined in the analysis of a steady state heat transfer problem. Each results in a modification of the linear equation generated at or near the boundary node.

a. The Dirichlet Boundary Condition

Figure 2.4 depicts the Dirichlet condition in the corner of a two dimensional region. The temperature profile is specified on the boundary as a constant or as a function. No discretization equation is required on boundary nodes since the temperature is already given by the Dirichlet condition. However, the discretization equation generated for a node adjacent to the boundary is affected. For example, incorporating the western boundary condition, g(y), at Node 9 in Figure 2.4 yields

$$g(y_{j+2}) + u_S - 4u_C + u_E + u_N = -\frac{\dot{q}h^2}{k}, \qquad (2.14)$$

which is rewritten as

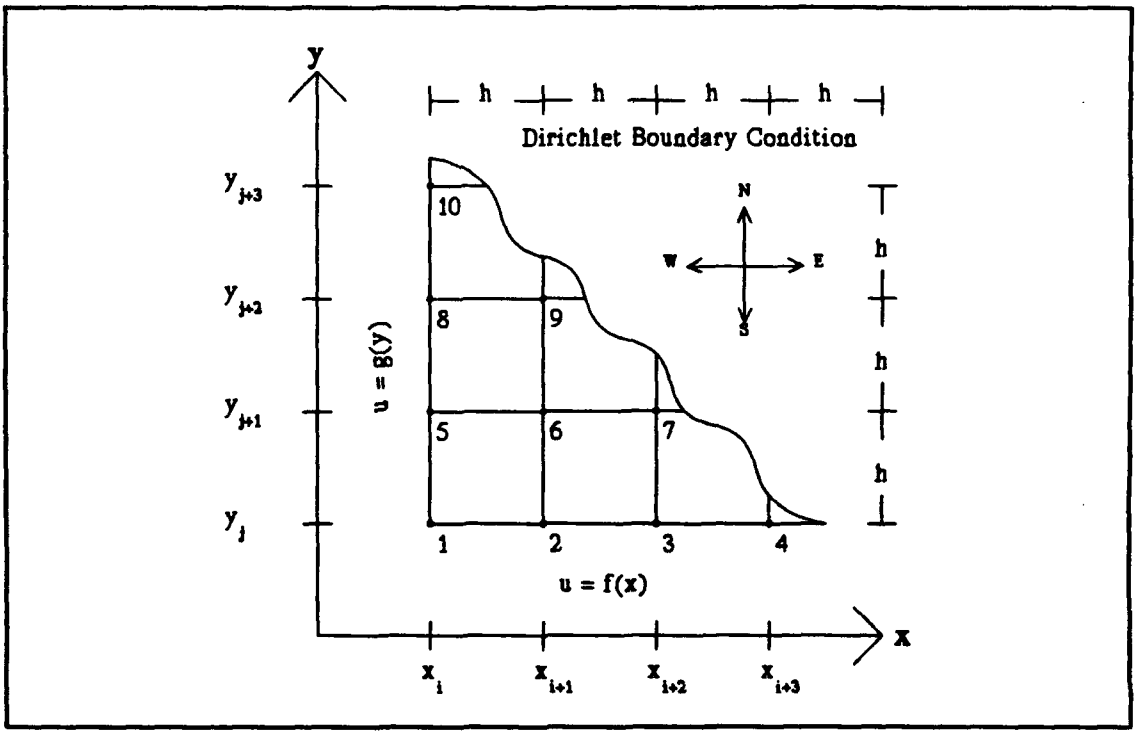


Figure 2.4: Dirichlet boundary condition in the corner of a two dimensional rectangular region.

$$3u_S - 4u_C + u_E + u_N = -\left(\frac{\dot{q}h^2}{k} + g(y_{j+2})\right). \tag{2.15}$$

Node 6 in Figure 2.4 is adjacent to two boundaries toward the west and south. At this node Equation 2.10 becomes

$$-4u_C + u_E + u_N = -\left(\frac{\dot{q}h^2}{k} + f(x_{i+1}) + g(y_{j+1})\right). \tag{2.16}$$

This analysis is similarly applied to the northern and eastern boundaries and is easily extended to three dimensions.

b. The Neumann Boundary Condition

The second major class of boundary conditions, the Neumann condition, is illustrated in Figure 2.5. For the Neumann boundary condition the magnitude of the outward pointing normal derivative is given along the boundary. Physically this derivative models the outward heat flux in the direction normal to the boundary. At Node 5 in Figure 2.5, Equation 2.10 is used to generate a difference equation. The term u_W appears in the equation but does not lie in the region. Hence the western node is fictitious. The non-existent u_W term is eliminated by using the difference equation that approximates the boundary condition which is given as

$$k\frac{\partial u}{\partial n} = g(y). \tag{2.17}$$

The derivatives in the direction of the outward pointing normal on the western, southern, and lower boundaries are the negative of the derivative taken in the direction positive to the respective axis. On the eastern, northern, and upper boundaries both derivatives point in the same direction. This subtle difference between the definitions of the two types of derivatives affects the form of the boundary condition approximations at the respective boundaries. The three term Taylor series expansion at Node 6 and at the fictitious western node produces respectively

$$u_W = u_C - h\frac{\partial u}{\partial x}\Big|_C + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}\Big|_C + O(h^3), \tag{2.18a}$$

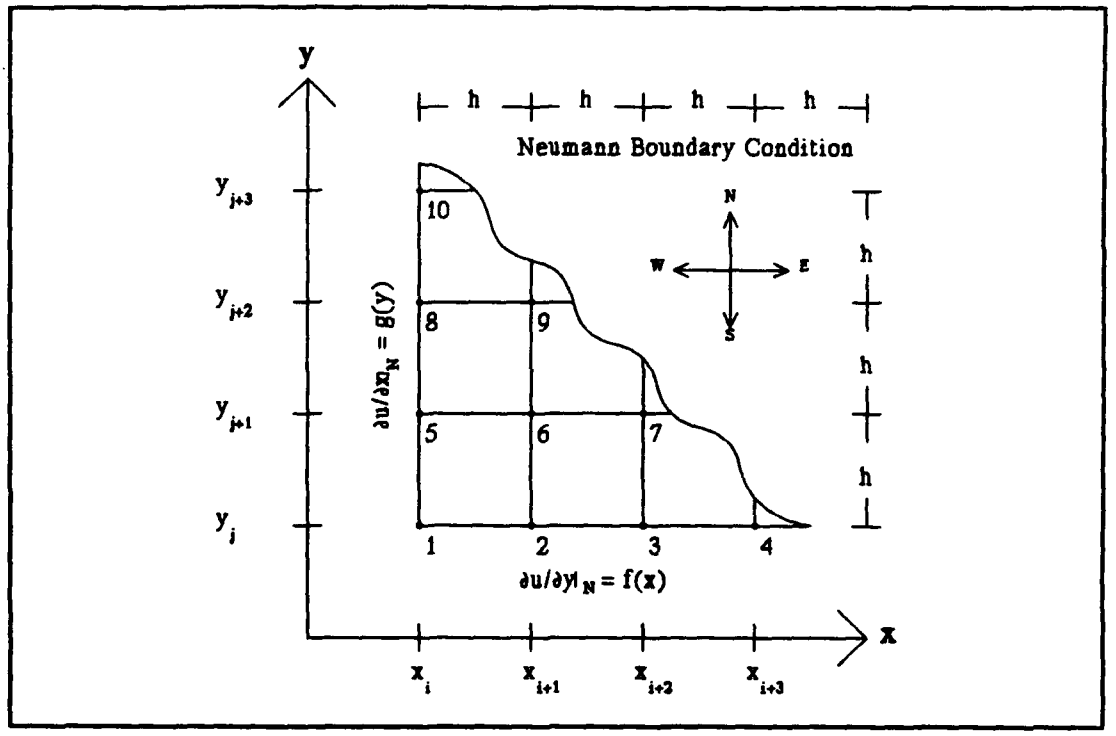


Figure 2.5: Neumann boundary condition on the corner of a two dimensional region.

$$u_E = u_C + h\frac{\partial u}{\partial x}\Big|_C + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}\Big|_C + O(h^3). \tag{2.18b}$$

Approximating the first partial derivative and taking its negative gives the discretization of the derivative in the direction of the outward pointing normal at the western boundary:

$$\frac{\partial u}{\partial n} = \frac{u_W - u_E}{2h} + O(h^2). \tag{2.19}$$

The difference equation for the Neumann boundary condition at Node 5 is

$$\frac{u_W - u_E}{2h} = \frac{g(y_{j+1})}{k}. \tag{2.20}$$

Solving for u_W gives

$$u_W = u_E + \frac{2hg(y_{j+1})}{k}. \tag{2.21}$$

The expression for u_W is directly substituted into Equation 2.10 giving

$$u_S - 4u_C + 2u_E + u_N = -\frac{1}{k}\left(\dot{q}h^2 + 2hg(y_{j+1})\right). \tag{2.22}$$

The order of the error after incorporating the Neumann condition is still h^2. This analysis is extended to all nodes lying on a Neumann boundary of a rectangular region if the grid is uniform.

c. *The Mixed Boundary Condition*

The mixed boundary condition, illustrated in Figure 2.6, incorporates both heat flux and temperature at the boundary. The equation governing this boundary condition is

$$Au + B\frac{\partial u}{\partial n} = g(y), \tag{2.24}$$

where A and B are functions defined on the region. The application of Equation 2.10 to Node 5 results in the inclusion of point u_W which does not lie in the region. This point is eliminated by writing the difference equation for the mixed boundary condition using Equation 2.19. This gives

$$Au_C + B\left(\frac{u_W - u_E}{2h}\right) = g(y), \tag{2.25}$$

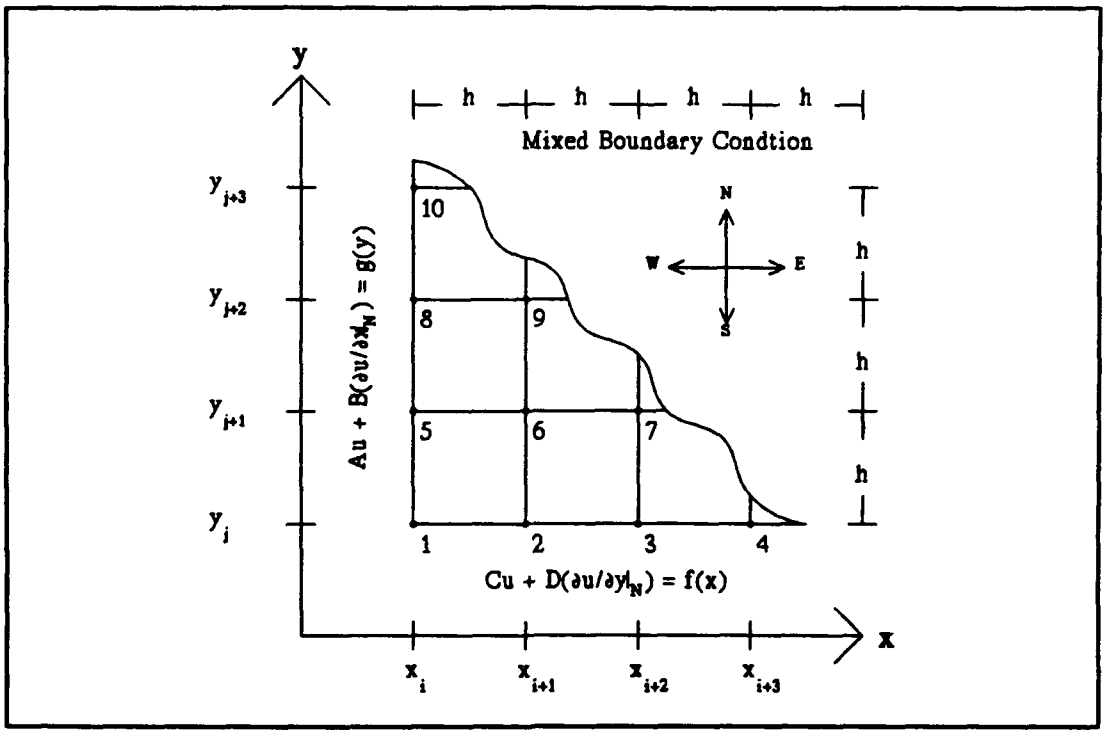


Figure 2.6: Mixed boundary conditions on the corner of a two dimensional region.

which when solved for u_W becomes

$$u_W = -\frac{2Ah}{B}u_C + u_E + \frac{2h}{B}g(y). \tag{2.26}$$

Substituting this expression into Equation 2.10 generates

$$u_S - (4 + \frac{2Ah}{B})u_C + 2u_E + u_N = -\left(\frac{\dot{q}h^2}{k} + \frac{2h}{B}g(y)\right). \tag{2.27}$$

The mixed boundary condition applies when convection is the mode of heat transfer at the boundary. Convection is modeled by Newton's Law of Cooling:

$$k\frac{\partial u}{\partial x}\Big|_n = H[u_\infty - u(x_i)], \tag{2.28}$$

where

- H = the convection coefficient.
- u_∞ = the temperature of the surrounding medium.

Equation 2.28 can be rewritten as

$$Hu(x_i) + k\frac{\partial u}{\partial n} = Hu_\infty \ . \qquad (2.29)$$

Equation 2.29 is a form of the mixed boundary condition stated in Equation 2.25 where A equals H, B equals k, and g(y) equals Hu_∞.

5. Example: Solving the Heat Equation on a Two Dimensional Chip Using a Uniform Grid

a. Problem Statement

In this problem the data provided in the following subsection is used to approximate the temperature profile in the two dimensional chip depicted in Figure 2.7. A uniform grid of interval length one millimeter is used to accomplish this approximation.

b. Data

A rectangular chip, shown in Figure 2.7, has dimensions of four millimeters by three millimeters. The western boundary of the chip is held at a constant 100°C (e.g. a Dirichlet boundary condition). The northern and southern boundaries are covered by a perfect insulator (e.g. a Neumann boundary condition with $u_n = 0$). The eastern boundary convects to the surrounding medium (e.g. a mixed boundary condition).

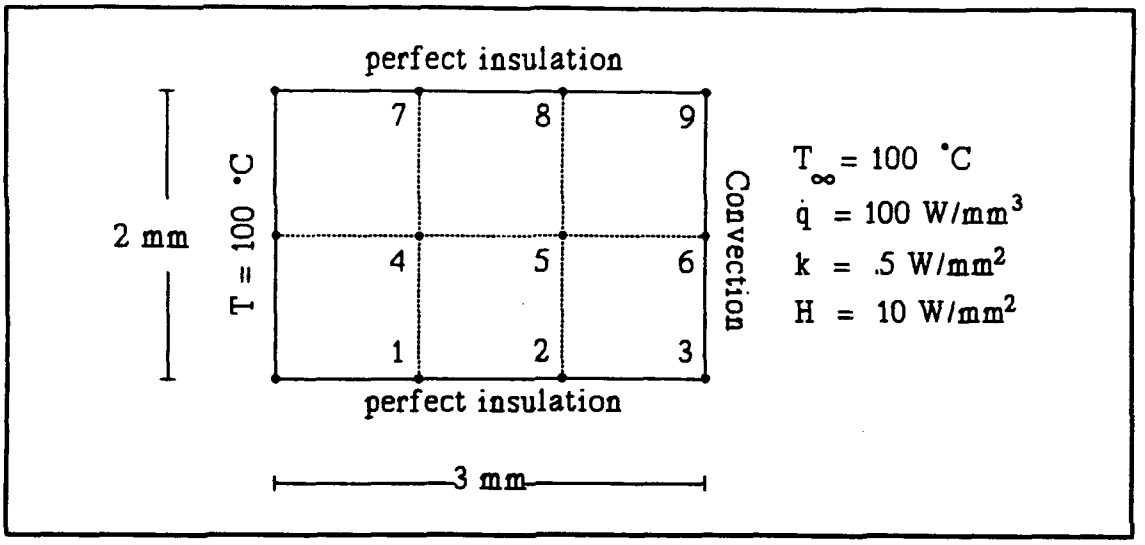


Figure 2.7: Set up for two dimensional uniform grid example.

The surrounding medium has a temperature of 100°C. A heat source of 100 watts/millimeter^3 is present inside the chip. The thermal conductivity coefficient for the chip material is equal to .5 watts/millimeter^2. The convection coefficient at the eastern boundary is 10 watts/millimeter^2.

*c. **Analysis***

In Figure 2.7 a nodal network is illustrated in which only the nodes of unknown temperature are listed. The temperature is given along the western boundary and therefore the nodes there are not numbered. The perfect insulator on the north boundary prevents the flow of heat in the direction normal to the boundary. The expressions for the boundary conditions are

$$\begin{aligned} &\text{east boundary:} && u = 100, \\ &\text{west boundary:} && 20u + \frac{\partial u}{\partial n} = 2000, \\ &\text{north/south boundary:} && \frac{\partial u}{\partial n} = 0. \end{aligned} \tag{2.30}$$

Node 5 is the only node whose difference equation is unaffected by the boundary conditions. Its equation is obtained by invoking Equation 2.10

$$u_2 + u_4 - 4u_5 + u_6 + u_8 = -200. \tag{2.31a}$$

Node 4 is affected by the western Dirichlet condition. The difference equation here is of similar form to that given in Equation 2.15

$$u_1 - 4u_4 + u_5 + u_7 = -300. \tag{2.31b}$$

Nodes 2 and 8 are affected by the Neumann boundary condition. Their equations respectively are

$$\begin{aligned} u_1 - 4u_2 + u_3 + 2u_5 &= -200, \\ 2u_6 + u_7 - 4u_8 + u_9 &= -200. \end{aligned} \tag{2.31c}$$

Equation 2.31c is of the form given in Equation 2.22 after it has been modified for the southern and northern boundaries, respectively.

Node 6 depends on a mixed boundary condition. Its equation is of the form given by Equation 2.27

$$u_3 + 2u_5 - 44u_6 + u_9 = -4200. \tag{2.31d}$$

The remaining nodes lie on the intersection of two boundaries. A combination of the previously discussed equations is used to generate the difference equations at these nodes. They are

$$\begin{aligned} -4u_1 + u_2 + 2u_4 &= -300, \\ 2u_2 - 44u_3 + 2u_6 &= -4200, \\ 2u_4 - 4u_7 + u_8 &= -300, \\ 2u_6 + 2u_8 - 44u_9 &= -4200. \end{aligned} \tag{2.31e}$$

d. Solution

Expressing Equation 2.31 in matrix form generates a sparse matrix that is banded about the main diagonal

$$\begin{bmatrix} -4 & 1 & & 2 & & & & & \\ 1 & -4 & 1 & & 2 & & & & \\ & 2 & -44 & & & 2 & & & \\ 1 & & & -4 & 1 & & 1 & & \\ & 1 & & 1 & -4 & 1 & & 1 & \\ & & 1 & & 2 & -44 & & & 1 \\ & & & 2 & & & -4 & 1 & \\ & & & & 2 & & 1 & -4 & 1 \\ & & & & & 2 & & 2 & -44 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix} = \begin{bmatrix} -300 \\ -200 \\ -4200 \\ -300 \\ -200 \\ -4200 \\ -300 \\ -200 \\ -4200 \end{bmatrix}. \tag{2.32}$$

It is now possible to invoke any of a number of methods to solve the system. The solution represents the approximate temperature at each node of the grid with an error of

order h^2 and is summarized in Figure 2.8. PC-MATLAB™, which uses Gaussian elimination, was used to solve this system [Ref. 3:p 3-191].

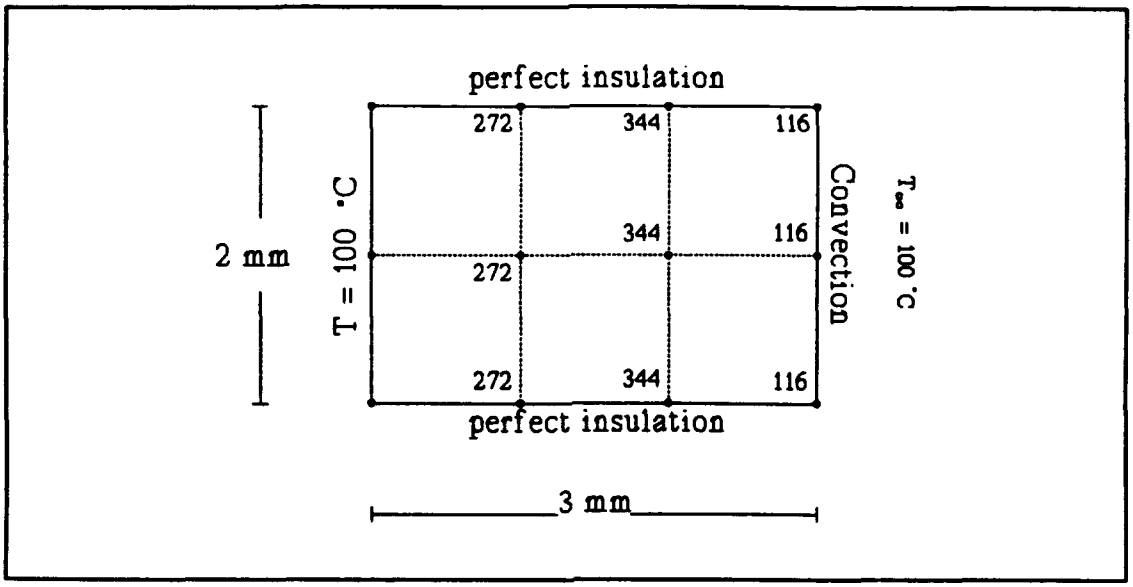


Figure 2.8: Example solution rounded to the nearest degree Celsius.

B. FINITE DIFFERENCE APPROXIMATIONS ON A NON-UNIFORM GRID.

1. Three Dimensional Analysis

Sometimes greater detail in the temperature profile is required for a specific portion of the region. This could be accomplished by refining the grid for the entire region. However this rapidly increases the number of nodes in the region and the corresponding size of the linear system of equations that must be solved. To avoid uncontrolled growth in the system of equations, the grid is refined only in the parts of the region where greater detail is desired. Figure 2.9 illustrates a nodal network in which the greatest detail is obtained in the center of the region by specifying a smaller interval

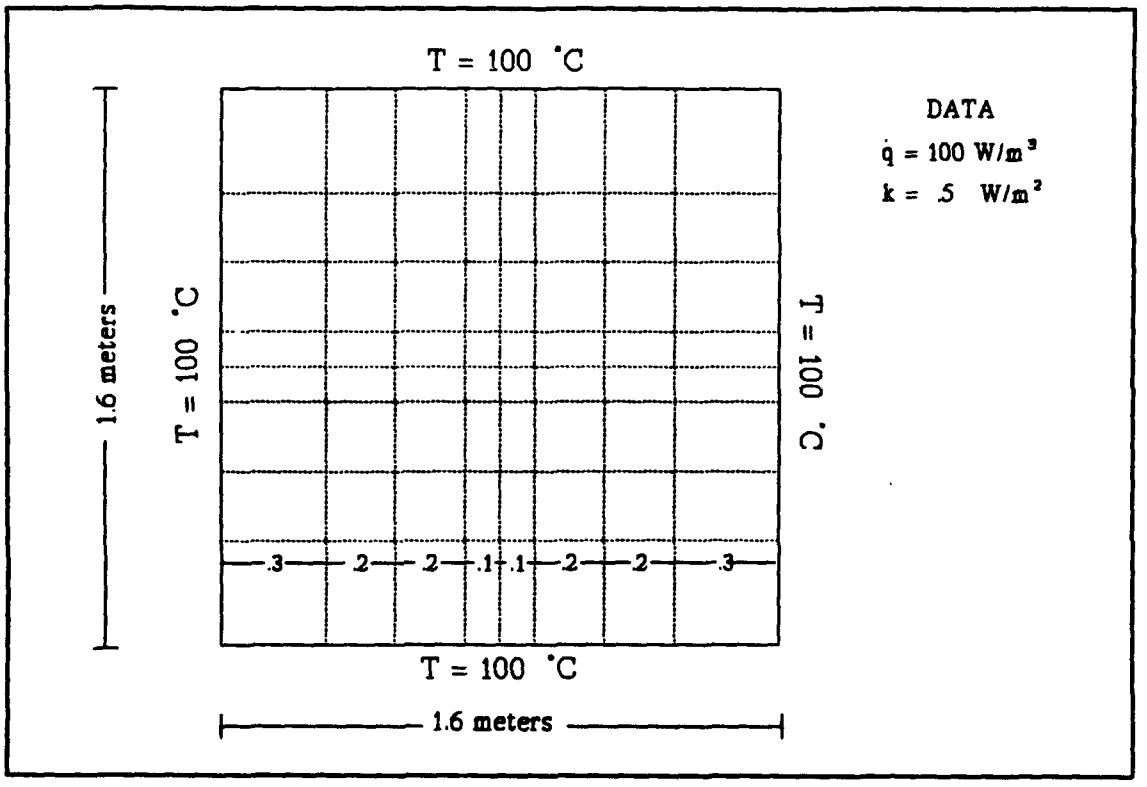


Figure 2.9: Non-uniform grid super-imposed onto a square region.

there. The resulting grid is non-uniform. Unfortunately, equations developed in the previous section can only be used for uniform grid analysis. A more general set of equations must be derived for the non-uniform grid.

A typical nodal point for a non-uniform grid in a three dimensional region is depicted is Figure 2.10 with the following terms defined

$$
\begin{aligned}
u_C &= u_{i,j,k} &&= u(x_i, y_j, z_k),\\
u_W &= u_{i-1,j,k} &&= u(x_i - \Delta x, y_j, z_k),\\
u_S &= u_{i,j-1,k} &&= u(x_i, y_j - \Delta y, z_k),\\
u_L &= u_{i,j,k-1} &&= u(x_i, y_j, z_k - \Delta z),\\
u_E &= u_{i+\alpha,j,k} &&= u(x_i + \alpha\Delta x, y_j, z_k),\\
u_N &= u_{i,j+\beta,k} &&= u(x_i, y_j + \beta\Delta y, z_k),\\
u_U &= u_{i,j,k+\gamma} &&= u(x_i, y_j, z_k + \gamma\Delta z).
\end{aligned}
\tag{2.33}
$$

where u(x,y,z) is temperature function for the region.

The difference approximations for the first partial derivatives on a non-uniform grid are obtained from the three term Taylor series expansions about the off-center nodes

$$
\begin{aligned}
u_W &= u_C - \Delta x\frac{\partial u}{\partial x}\Big|_C + (\Delta x)^2\frac{\partial^2 u}{\partial x^2}\Big|_C + O\left[(\Delta x)^3\right],\\
u_E &= u_C + \alpha\Delta x\frac{\partial u}{\partial x}\Big|_C + (\alpha\Delta x)^2\frac{\partial^2 u}{\partial x^2}\Big|_C + O\left[(\Delta x)^3\right],
\end{aligned}
\tag{2.34a}
$$

$$
\begin{aligned}
u_S &= u_C - \Delta y\frac{\partial u}{\partial y}\Big|_C + (\Delta y)^2\frac{\partial^2 u}{\partial y^2}\Big|_C + O\left[(\Delta y)^3\right],\\
u_N &= u_C + \beta\Delta y\frac{\partial u}{\partial y}\Big|_C + (\beta\Delta y)^2\frac{\partial^2 u}{\partial y^2}\Big|_C + O\left[(\Delta y)^3\right],
\end{aligned}
\tag{2.34b}
$$

$$
\begin{aligned}
u_L &= u_C - \Delta z\frac{\partial u}{\partial z}\Big|_C + (\Delta z)^2\frac{\partial^2 u}{\partial z^2}\Big|_C + O\left[(\Delta z)^3\right],\\
u_U &= u_C + \gamma\Delta z\frac{\partial u}{\partial z}\Big|_C + (\gamma\Delta z)^2\frac{\partial^2 u}{\partial z^2}\Big|_C + O\left[(\Delta z)^3\right].
\end{aligned}
\tag{2.34c}
$$

From Equation 2.34 the first derivative approximations are shown to be

$$\frac{\partial u}{\partial x}\Big|_C = \frac{u_E + (\alpha^2 - 1)u_C - \alpha^2 u_W}{\alpha(\alpha + 1)\Delta x} + O\left[(\Delta x)^2\right], \tag{2.35a}$$

$$\frac{\partial u}{\partial y}\Big|_C = \frac{u_N + (\beta^2 - 1)u_C - \beta^2 u_S}{\beta(\beta + 1)\Delta y} + O\left[(\Delta y)^2\right], \tag{2.35b}$$

$$\frac{\partial u}{\partial z}\Big|_C = \frac{u_U + (\gamma^2 - 1)u_C - \gamma^2 u_L}{\gamma(\gamma + 1)\Delta z} + O\left[(\Delta z)^2\right]. \tag{2.35c}$$

Equation 2.35 is used in the analysis of Neumann and mixed boundary conditions on the non-uniform grid.

The second derivative approximations are also obtained from Equation 2.34:

$$\frac{\partial^2 u}{\partial x^2}\Big|_C = \frac{2u_E - 2(\alpha + 1)u_C + 2\alpha u_W}{\alpha(\alpha + 1)\Delta^2 x} + O(\Delta x), \tag{2.36a}$$

$$\frac{\partial^2 u}{\partial y^2}\Big|_C = \frac{2u_N - 2(\beta + 1)u_C + 2\beta u_S}{\beta(\beta + 1)\Delta^2 y} + O(\Delta y), \tag{2.36b}$$

$$\frac{\partial^2 u}{\partial z^2}\Big|_C = \frac{2u_U - 2(\gamma + 1)u_C + 2\gamma u_L}{\gamma(\gamma + 1)\Delta^2 z} + O(\Delta z). \tag{2.36c}$$

Equation 2.36 is a first order approximation (i.e the order of the error is h) to the second derivative of the temperature function u(x,y,z). In contrast, the uniform grid counterpart of Equation 2.39 given by Equation 2.9 is a second order approximation (i.e. the order of the error is h^2) of the second derivative of the temperature function.

Using Equation 2.36, the approximation to Poisson's Equation for heat transport on a non-uniform grid becomes

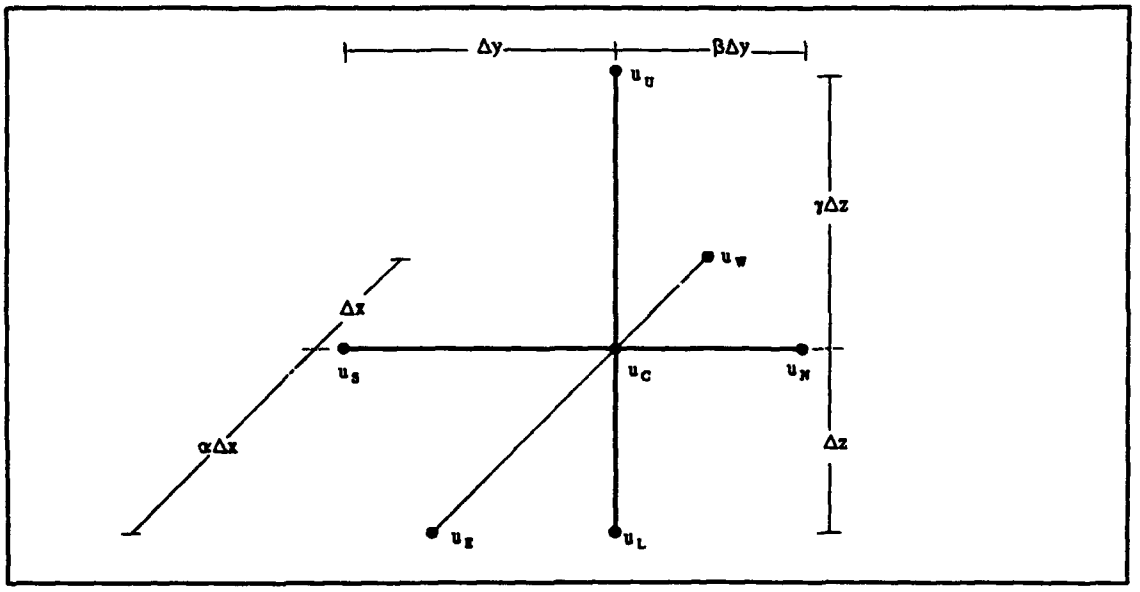


Figure 2.10: A typical nodal network on a non-uniform grid in three dimensions.

$$\frac{u_E - (\alpha+1)u_C + \alpha u_W}{\alpha(\alpha+1)\Delta^2 x} + \frac{u_N - (\beta+1)u_C + \beta u_S}{\beta(\beta+1)\Delta^2 y} + \frac{u_U - (\gamma+1)u_C + \gamma u_L}{\gamma(\gamma+1)\Delta^2 z} = -\frac{\dot{q}}{2k} . \tag{2.37}$$

For convenience this is rewritten as

$$\frac{1}{\alpha(\alpha+1)\Delta^2 x}u_E + \frac{1}{\beta(\beta+1)\Delta^2 y}u_N + \frac{1}{\gamma(\gamma+1)\Delta^2 z}u_U - \left(\frac{1}{\alpha\Delta^2 x} + \frac{1}{\beta\Delta^2 y} + \frac{1}{\gamma\Delta^2 z}\right)u_C + \frac{1}{(\alpha+1)\Delta^2 x}u_W + \frac{1}{(\beta+1)\Delta^2 y}u_S + \frac{1}{(\gamma+1)\Delta^2 z}u_L = -\frac{\dot{q}}{2k} . \tag{2.38}$$

Equation 2.38 is applied to each node to generate the system of equations whose solution approximates the temperature profile on a non-uniform grid in a rectangular region.

2. Boundary Conditions

Boundary conditions for the non-uniform grid are handled in the same way as they were when a uniform grid is used. If a Dirichlet boundary condition is present, the value of the temperature of the boundary is substituted for the appropriate node in Equation 2.38.

If a Neumann or mixed boundary condition is employed, the difference equation approximations specified in Equations 2.20 and 2.25, respectively are used and solutions for the fictitious node are found. A fictitious node is chosen to be at the same distance from the boundary as its existent counterpart on the opposite side. This allows uniform grid analysis on the Neumann or mixed boundary points. The expressions due to a fictitious node from the Neumann condition (Equation 2.21) or from the mixed condition (Equation 2.26) are substituted for the appropriate node in Equation 2.37.

3. Example: Solving the Heat Equation on a Two Dimensional Chip Using a Non-Uniform Grid.

a. Problem Statement

In this example the temperature profile is approximated for a two dimensional chip with dimension of 4 millimeters by 4.5 millimeters using the non-uniform grid, boundary conditions, and data specified in Figure 2.11.

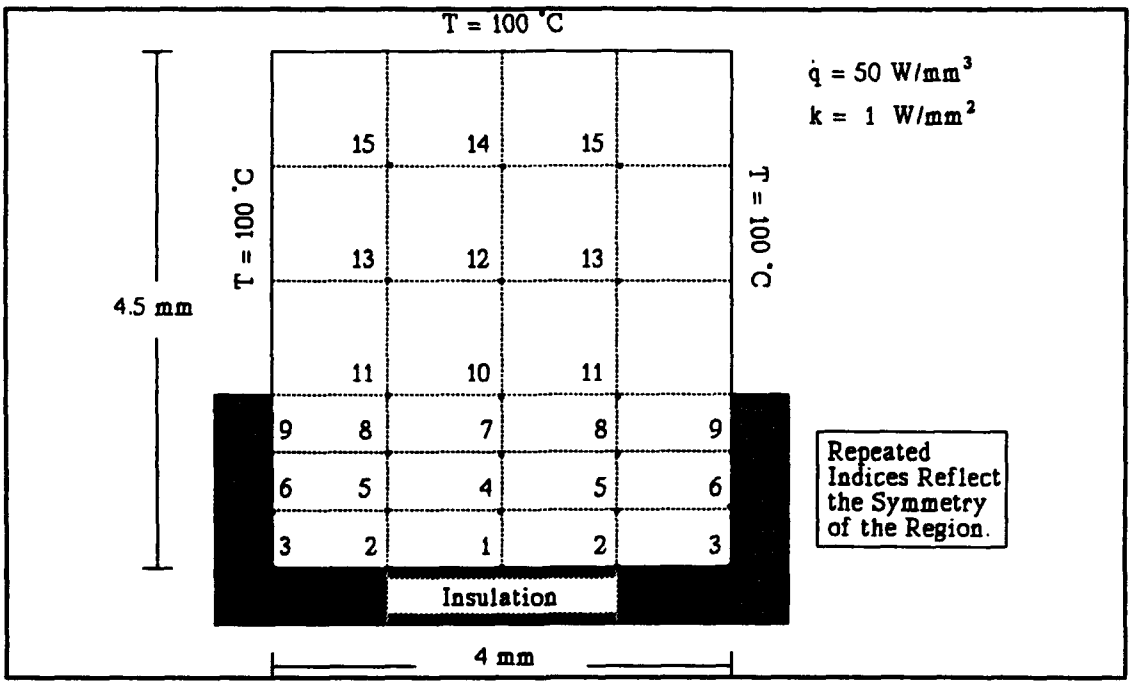


Figure 2.11: Set-up for non-uniform grid example.

b. Analysis

The number of equations in the resulting linear system is reduced by using the symmetry of the region. The numbering scheme of the nodes in Figure 2.11 reflects this symmetry. Throughout most of the region, the grid is uniform and Equation 2.10 is used to generate the difference equations that will approximate the temperature profile. However, when Node 10 or Node 11 is in the center of the nodal network, e.g. $u_C = u_{10}$ or $u_C = u_{11}$, there is non-uniformity in the y direction. At these nodes the grid length in the y direction is 1/2 millimeter to the south and one millimeter to the north and therefore $\beta = 2$. The grid is still uniform in the x direction at Nodes 10 and 11 and therefore $\alpha = 1$. Utilizing Equation 2.38, modified for two dimensions, and incorporating the boundary conditions generates the following sparse matrix system

$$\begin{bmatrix}
-10 & 2 & & 8 & & & & & & & & & & & \\
1 & -10 & 1 & & 8 & & & & & & & & & & \\
 & 1 & -5 & & & 4 & & & & & & & & & \\
2 & & & -5 & 1 & & 2 & & & & & & & & \\
 & 4 & & 1 & -10 & 1 & & 4 & & & & & & & \\
 & & 2 & & 1 & -5 & & & 2 & & & & & & \\
 & & & 2 & & & -5 & 1 & & 2 & & & & & \\
 & & & & 4 & & 1 & -10 & 1 & & 4 & & & & \\
 & & & & & 2 & & 1 & -5 & & & & & & \\
 & & & & & & 4 & & & -9 & 3 & 2 & & & \\
 & & & & & & & 8 & & 3 & -18 & & 4 & & \\
 & & & & & & & & & 1 & & -4 & 2 & 1 & \\
 & & & & & & & & & & 1 & 1 & -4 & & 1 \\
 & & & & & & & & & & & 1 & & -4 & 2 \\
 & & & & & & & & & & & & 1 & 1 & -4
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \\ u_{15} \end{bmatrix}
=
\begin{bmatrix} -50 \\ -50 \\ -25 \\ -25 \\ -50 \\ -25 \\ -25 \\ -50 \\ -225 \\ -75 \\ -450 \\ -50 \\ -150 \\ -150 \\ -150 \end{bmatrix}. \quad (2.39)$$

c. Solution

The solution to Equation 2.39 is summarized in Figure 2.12. As in the previous example, PC-MATLAB™ was used to solve the matrix system. The results are rounded to the nearest degree Celsius.

d. Error of the Approximation

Recall that the error in the approximation when using the non-uniform grid is given by

$$\text{Error} = K \times \text{Length of Sub-interval.}$$

Since the mesh spacing is non-uniform, the value that represents the longest interval length of the grid is used for the length of the sub-interval.

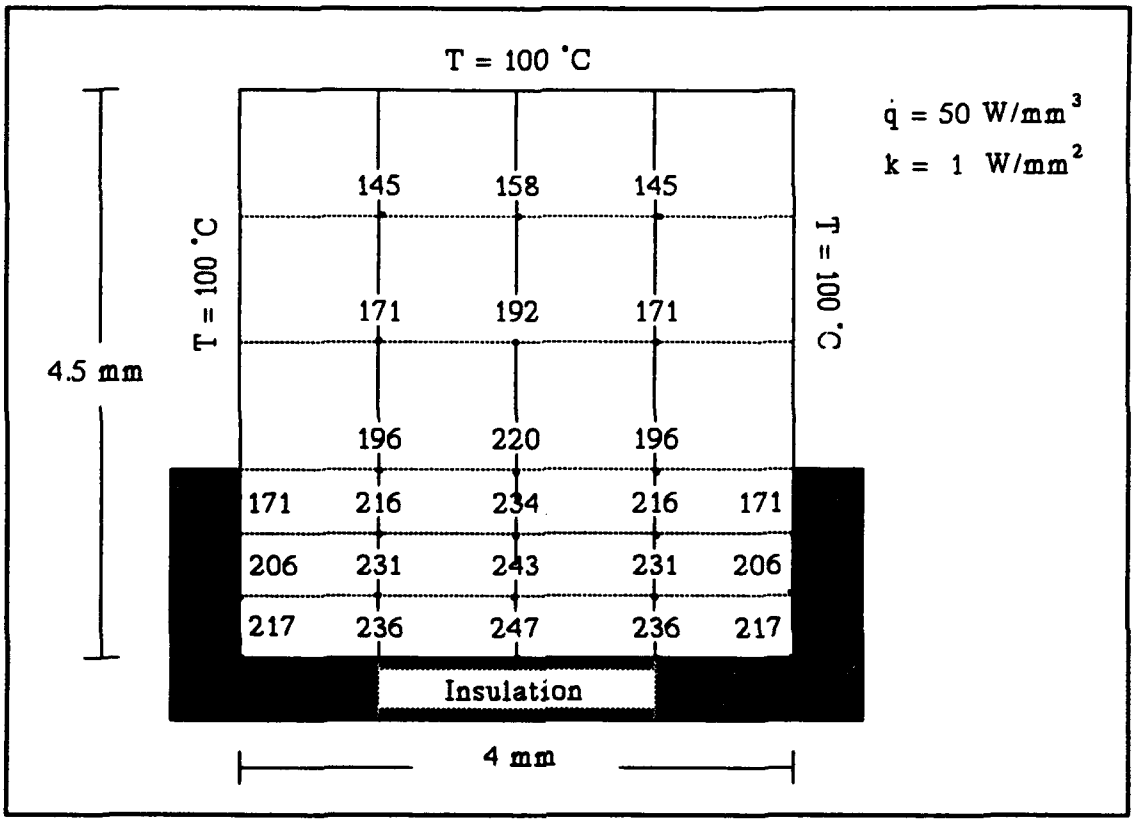


Figure 2.12: Solution to non-uniform grid example with temperatures rounded to the nearest degree Celsius.

This example is revisited in Chapter III when the use of ELLPACK is demonstrated.

III. SOLVING ELLIPTIC PROBLEMS USING ELLPACK

ELLPACK is a multi-facetted system used to solve elliptic partial differential equations. The ELLPACK project originated in 1976 at Purdue University under the direction of John R. Rice. It continues under development today but, as it stands, is more advanced than what is available in current FORTRAN libraries. ELLPACK gives an easy way to solve simple to moderately complex elliptic equations. With a little more work ELLPACK can be used to solve even harder problems. In order to use ELLPACK to its fullest potential a fundamental understanding of FORTRAN is necessary. A detailed discussion of ELLPACK is found in Rice and Boisvert [Ref. 4]. This chapter will serve as a primer to the use of ELLPACK and will provide those tools necessary to solve the problem posed in Chapter I.

A. CLASSIFYING PARTIAL DIFFERENTIAL EQUATIONS

In order to discuss the capabilities and limitations of ELLPACK the means of classifying partial differential equations must be understood.

1. The Elliptic Equation.

The most general form of a linear second order partial differential equation is given by

$$au_{xx} + 2bu_{xy} + cu_{yy} + du_x + eu_y + fu = g, \qquad (3.1)$$

where a, b, c, d, e, f, and g are functions of x and y; u_x and u_y are first partial derivatives of the function u(x,y); and u_{xx}, u_{xy}, and u_{yy} are second partial derivatives of u(x,y). An equation is said to be elliptic if

$$b^2 - ac < 0 \tag{3.2}$$

for all x and y in the region for which the equation is defined.

If a = c = 1 and b = d = e = f = 0, Equation 3.1 becomes

$$u_{xx} + u_{yy} = g, \tag{3.3}$$

which is Poisson's equation. Equation 3.3 is further specialized by letting g = 0. The result is

$$u_{xx} + u_{yy} = 0, \tag{3.4}$$

which is Laplace's equation. In Equations 3.3 and 3.4 the value for b^2-ac is -1. Hence Poisson's equation and Laplace's equation are both elliptic.

2. The Self-Adjoint Equation

The property of self-adjointness is also used to classify partial differential equations. Without going into a great amount of detail, Equation 3.1 is said to be self-adjoint if

where a, b, c, d, e, f, and g are functions of x and y; u_x and u_y are first partial derivatives of the function u(x,y); and u_{xx}, u_{xy}, and u_{yy} are second partial derivatives of u(x,y). An equation is said to be elliptic if

$$b^2 - ac < 0 \tag{3.2}$$

for all x and y in the region for which the equation is defined.

If a = c = 1 and b = d = e = f = 0, Equation 3.1 becomes

$$u_{xx} + u_{yy} = g, \tag{3.3}$$

which is Poisson's equation. Equation 3.3 is further specialized by letting g = 0. The result is

$$u_{xx} + u_{yy} = 0, \tag{3.4}$$

which is Laplace's equation. In Equations 3.3 and 3.4 the value for b^2-ac is -1. Hence Poisson's equation and Laplace's equation are both elliptic.

2. The Self-Adjoint Equation

The property of self-adjointness is also used to classify partial differential equations. Without going into a great amount of detail, Equation 3.1 is said to be self-adjoint if

$$pu_n + qu = r(x,y), \tag{3.9}$$

where p and q are constant along each side of the region. Recall that u_n is the direction of the outward pointing normal. If a problem is separable, it may be solved using very fast and efficient solution techniques.

4. Boundary Conditions

Partial differential equations are also classified by boundary condition type as discussed in Chapter II. In brief review, if p is equal to zero in Equation 3.9 then the boundary conditions are of Dirichlet type. If q is equal to zero in Equation 3.9 then the boundary conditions are of Neumann type. If neither p nor q are equal to zero in Equation 3.9 it is called a mixed boundary condition. A region may have any combination of these boundary conditions.

In summary, the equations discussed in Chapter I that are used to model heat transport are second order, linear, and self-adjoint. If either the Dirichlet, Neumann, or mixed boundary conditions are prescribed, and if the region under investigation is rectangular then these equations are also separable. The categories of classification described in this section are not exhaustive, but they are sufficient to continue the discussion of the problem posed in this paper. It should be noted however, that once mastered, ELLPACK can also be used to solve parabolic, non-linear, and other non-standard problems. For more information see Rice and Boisvert [Ref. 4:pp 87-135].

B. MATRIX PROPERTIES

The matrices that result from the discretization of elliptic partial differential equations have several desirable properties that numerical routines use to their advantage.

1. Sparsity

When using the Seven Point Star method to approximate the temperature profile in a three dimensional regime there are at most seven non-zero elements per equation. Hence an n by n matrix will have at least n^2-7n zero elements. This preponderance of zeroes is known as sparsity.

2. Bandedness

In the examples solved in Chapter II it was seen that the non-zero elements of the matrix are confined to bands that lie on either side of the main diagonal. The bandwidth for a matrix A is defined as:

$$\max \; |i-j| \text{ for which } A_{i,j} \neq 0.$$

3. Symmetry

Matrix A is said to be symmetric if $A_{i,j} = A_{j,i}$. In general, elliptic problems lead to symmetric matrices if the region is symmetric about its center. For example, Figure 2.9 poses a problem whose linear system will result in a symmetric matrix.

4. Positive Definiteness

Matrix A is said to be positive definite if $x^TAx > 0$ for all non-zero x vectors. The eigenvalues for symmetric positive definite matrices are real and positive.

C. THE STRUCTURE OF ELLPACK

ELLPACK is an extension of FORTRAN, therefore FORTRAN statements can be mixed with ELLPACK statements. A basic block of statements in ELLPACK is called a segment. Segments contain statements that are used to define the equations, domains, and boundary conditions. Segments can also invoke powerful modules in the ELLPACK library that contain the routines that will solve the problem. Basic types of ELLPACK modules include:

- Discretization modules which create discrete approximations to the elliptic problem.
- Indexing modules which order the linear equations and unknowns generated by the discretization module.
- Solution modules which solve the system of linear equations.
- Output modules which specify ELLPACK generated output.

Once a program is executed in ELLPACK it is converted by the ELLPACK preprocessor into a FORTRAN program. At this point the program is compiled. At execution, the program invokes modules from the ELLPACK library specified in the segment blocks. These modules are also written in FORTRAN. Output can be either printed or plotted using ELLPACK routines. Output may also be displayed using any means available in existing FORTRAN libraries.

D. ELLPACK SEGMENTS

ELLPACK segments fall into four basic groups. We describe some of the segments below. Those described are used to model the problem posed in Chapter I. Many more segments not described here are available in the ELLPACK system and are discussed in detail in Rice and Boisvert [Ref. 4:pp. 24-44].

1. Group One Segments

Group one segments define the elliptic problem and must appear before any group two segment. Group one segments include

a. EQUATION Segment

The EQUATION segment defines the partial differential equation. This segment can only be listed once in the program. The most general form of the EQUATION statement for a problem in two dimensions is

$$\begin{aligned} &\pm\langle\text{expression}\rangle * UXX \pm \langle\text{expression}\rangle * UXY \pm \langle\text{expression}\rangle * UYY \;\&\\ &\pm\langle\text{expression}\rangle * UX \pm \langle\text{expression}\rangle * UY \pm \langle\text{expression}\rangle * U \;\& \\ &\qquad = \langle\text{expression}\rangle, \end{aligned} \tag{3.10}$$

where <expression> represents any legal, real or double precision FORTRAN arithmetic expression or function. The ampersand at the end of the first and second lines in Expression 3.10 is an ELLPACK symbol that indicates that the expression continues on the next line. The self adjoint form of Expression 3.10 is

$$\pm(\langle\text{expression}\rangle * UX)X \pm (\langle\text{expression}\rangle * UY)Y \pm \langle\text{expression}\rangle * U \quad \& \quad = \langle\text{expression}\rangle. \tag{3.11}$$

By way of example, the three dimensional form of Poisson's equation for heat flow would appear in an ELLPACK EQUATION segment as

$$\text{EQUATION.} \qquad UXX + UYY + UZZ = -Q(X,Y,Z)/K(X,Y,Z), \tag{3.12}$$

where Q(X,Y,Z) and K(X,Y,Z) are functions describing the material properties over the region. Similarly, Laplace's equation would appear as

$$\text{EQUATION.} \qquad UXX + UYY + UZZ = 0. \tag{3.13}$$

An example of an equation segment written in self-adjoint form is

$$\text{EQUATION.} \quad (A(X,Y,Z)*UX)X + (B(X,Y,Z)*UY)Y \quad \& \quad + (C(X,Y,Z)*UZ)Z = 0. \tag{3.14}$$

where the coefficients A(X,Y,Z), B(X,Y,Z), and C(X,Y,Z) can vary or remain constant over the region.

The ELLPACK preprocessor does not perform an exhaustive check to ensure that the partial differential equation entered by the user in the EQUATION segment is elliptic. Furthermore, some of the modules require that the self-adjoint form

of the equation be used. It is the responsibility of the user to be aware of these requirements in order to effectively use the ELLPACK system.

b. BOUNDARY Segment

The BOUNDARY segment specifies the location of the boundaries of the region and the conditions on those boundaries. The expression for a boundary condition is broken up into two parts, e.g.

<condition> ON <piece>,

where condition is of the form

$$\pm\text{<expression>}*UX \pm \text{<expression>}*UY \pm \text{<expression>}*U \;\; \& \;\; = \text{<expression>} \tag{3.15}$$

and <piece> describes the location of the boundary. For example, a Dirichlet boundary condition where the temperature is 100° on an eastern boundary where x = 0 would appear as

BOUNDARY . U = 100 ON X = 0. (3.16)

A Neumann boundary condition where a perfect insulator is placed on the southern boundary where y = 0 would appear as

BOUNDARY . *UY* = 0 *ON* *Y* = 0. (3.17)

A mixed boundary condition modeling convection at z = 0 would have the form

$$\text{BOUNDARY .} \quad 10*UZ - U = -H*TINF \quad \text{ON} \quad Z = 0, \tag{3.18}$$

where TINF is the ambient temperature of the fluid region.

The preceding examples represent a fraction of ELLPACK's boundary assignment capabilities. Descriptions for defining complex boundary conditions are discussed in Rice and Boisvert [Ref. 4:pp. 29-32].

2. Group Two Segments

Group two segments specify executable ELLPACK modules and may appear more than once in the ELLPACK program. In general, they specify numerical methods to solve the elliptic problem and they must be placed in a specific order for the program to run properly, e.g. GRID, DISCRETIZATION, INDEXING, SOLUTION, and OUTPUT.

a. GRID Segment

The GRID segment defines a rectangular grid that is placed over the domain. The general form of a GRID segment statement is

<n> <variable> POINTS <point list>,

where

- <n> indicates the number of points,
- <variable> indicates the variable involved (X, Y, or Z),

- <point list> indicates the list of grid coordinates in increasing order.

For example one may specify

$$\begin{array}{lll} \text{GRID.} & \text{5 X POINTS} & 0.,\ .4,\ .5,\ .6,\ 1.0 \\ & \text{5 Y POINTS} & 0.,\ .3,\ .5,\ .7,\ 1.0 \end{array} \tag{3.19}$$

The result would be a rectangular grid with lines at

$$\begin{array}{lllll} x = 0.0, & x = .4, & x = .5, & x = .6, & x = 1.0, \\ y = 0.0, & y = .3, & y = .5, & y = .7, & y = 1.0. \end{array}$$

If a rectangular domain was specified by the BOUNDARY segment then the following grid could be defined in the following special form:

$$\begin{array}{lll} \text{GRID.} & \text{7 X POINTS} \\ & \text{7 Y POINTS} \end{array} \tag{3.20}$$

In Equation 3.20 the point list has been omitted. This causes ELLPACK to generate a uniform grid. More complex grid definitions are available and discussed in Rice and Boisvert [Ref. 4:pp. 32-35].

b. DISCRETIZATION Segment

The DISCRETIZATION segment names the module that is to be used to form the linear system of equations that will approximate the solution to the elliptic equation. The primary discretization modules used in this paper are:

(1) 5-POINT STAR Module This module generates difference equations using the ordinary second order central difference approximations that were discussed earlier. The following restrictions apply when using this module:

- The domain must be two dimensional.
- The domain must be rectangular.
- The grid size must be at least 3 by 3.
- There can be no u_{xy} terms in the equation.
- There can be no tangential derivative components in the boundary conditions.

(2) 7-POINT STAR Module This module is the three dimensional counterpart of the 5-POINT STAR module. However, the following additional restrictions apply:

- The region must be three dimensional.
- The equation must be self-adjoint (this implies that there may be no u_{xy}, u_{xz}, or u_{yz} terms in the equation).

c. *INDEXING Segment*

INDEXING modules reorganize the system of equations generated by the DISCRETIZATION segment by renumbering the equations and unknowns. If no module is specified in the main program then the ELLPACK system will invoke the AS IS module by default. Several other indexing modules are described in Rice & Boisvert [Ref. 4:p. 37], but the AS IS module is the only one used in this investigation.

(1) AS IS Module The AS IS module indexes the equations as they were generated by the discretization module. For example, if the nodes in a three dimensional region are designated by (i,j,k) at a point whose coordinates were (x_i, y_j, z_k), then the AS IS module would designate the index 1 to node (1,1,1), the index 2 to node (2,1,1), and so forth. If there are NX grid points defined in the x direction, then the node (1,2,1) is assigned the index NX+1. If NY grid points are defined in the y direction, then the node (1,1,2) is be assigned the index (NX)(NY) + 1. In general, the node (i,j,k) is assigned the index:

$$\text{Index Value} = i + NX*((j-1) + NY*(k-1)) \tag{3.21}$$

where '*' indicates multiplication in FORTRAN. The INDEXING segment need not appear in the ELLPACK program for the AS IS module to be invoked.

d. SOLUTION Segment

The SOLUTION segment specifies the module that will be used to solve the linear system that was generated by the DISCRETIZATION segment. Detailed information on ELLPACK solution modules is available in Rice and Boisvert [Ref. 4:pp. 178-217]. Below is a brief description of a few of the modules used for solving systems of linear equations.

(1) JACOBI SI (CG) Module This module uses iterative methods to solve a system of equations. Convergence is accelerated by using semi-iteration (conjugate gradient) techniques. Several optional parameters can be set and are discussed

in Rice and Boisvert [Ref. 4:pp. 183-184]. Some of these parameters are summarized in Table 3.1. If these parameters are not set, ELLPACK will use the default settings.

Table 3.1: Parameters/Default Settings for Jacobi SI(CG) and SOR Methods.

PARAMETER	MEANING OR USAGE	DEFAULT
ITMAX	Maximum number of iterations.	100
ZETA	Tolerance level in stopping test.	5.E-6
CME	Initial estimate of largest eigenvalue of the Jacobi matrix.	0.0
SME	Initial estimate of smallest eigenvalue of the Jacobi matrix for Jacobi SI method.	0.0
FF	Adaptive procedure damping factor.	0.75
OMEGA	Over relaxation parameter for SOR and SYMMETRIC SOR methods.	1.0
SPECR	Initial estimate of spectral radius of the SYMMETRIC SOR matrix.	0.0
BETAB	Initial estimate of the spectral radius of the LU matrix in SYMMETRIC SOR methods.	.25

(2) SOR Module This module solves a system of linear equations iteratively using over-relaxation techniques to accelerate the convergence. As with the JACOBI SI (CG), the module allows the user to select several parameters. If these are not set, the module automatically uses the default values indicated in Table 3.1.

(3) SPARSE LDLT Module This module solves a system of sparse linear equations by converting the equations to sparse storage and then using sparse Gauss elimination with column pivoting.

(4) BAND GE Module This module solves a real banded system of linear equations using LU factorization and Gaussian elimination with scaled partial pivoting.

e. OUTPUT Segment

The OUTPUT segment specifies various kinds of output generated from the computations. The basic form of the statement is

<type> (<function>),

where <type> is a keyword that prescribes the type of output desired and <function> is the function from which the output was generated. For example the statement PLOT(U) in the OUTPUT segment will provide a contour plot of the function U which is being used to describe the temperature field. The statement TABLE(U) will print a table of the function U at the grid points. Other possibilities are available and are described in detail in Rice and Boisvert [Ref. 4:pp. 43-44]. Output, however, is not limited to ELLPACK modules. Any output routines that are written by the user or that are available in external libraries may be invoked from FORTRAN segments.

3. Group Three Segments

Group three segments may occur anywhere in the program and as many times as are required. They include:

a. DECLARATIONS Segment

The DECLARATIONS segment allows the user to declare variables that will be used later in FORTRAN segments.

b. GLOBAL Segment

The GLOBAL segment is used to make variables generated by ELLPACK available for use in FORTRAN segments.

c. OPTIONS Segment

The OPTIONS segment sets various switches that control the ELLPACK program. A detailed list of options is provided in Rice and Boisvert [Ref. 4:pp. 41-42]. It is recommended that the OPTIONS segment be placed early in the program. OPTIONS segment modules available include:

(1) TIME Module Requests a printed summary of the execution times of each module.

(2) MEMORY Module Requests a printed summary of the memory used in the ELLPACK run.

(3) LEVEL = K Module Sets levels (0 - 5) of desired printed output. These levels are summarized in Table 3.2. If the level is not specified in the OPTIONS segment, the default (LEVEL = 2) will be set.

Table 3.2: Amount of Printed Material Invoked by LEVEL Module

LEVEL	DATA OUTPUT
1	Requests no output form modules except for fatal error messages.
2	Requests minimal output (default).
3, 4, 5	Greater Amounts of intermediate output. Used for debugging.

d. FORTRAN Segment

The FORTRAN segment contains lines of FORTRAN code provided by the user. It is the FORTRAN segment that gives the user the power to modify ELLPACK routines and obtain solutions to more difficult problems.

e. Miscellaneous Group Three Segments

Also considered as a group three segment is the symbol '*' and the blank line. The symbol '*' placed in column one of the code indicates a comment line. A blank line may be inserted anywhere in the ELLPACK program as desired by the user and has no effect on the program execution.

4. Group Four Segments

Group four segments must appear after all other segments of groups one, two, or three.

a. SUBPROGRAMS Segment

The SUBPROGRAMS segment provides a place for the user to insert complete FORTRAN functions and subroutines that are called from FORTRAN segments. This segment must be placed at the end of the ELLPACK program just before the END segment.

b. END Segment

The END segment specifies the end of the FORTRAN program.

E. SAMPLE ELLPACK PROGRAMS

This section uses the examples solved in Chapter II as models to build sample ELLPACK programs.

1. Sample Program One: Finding the Temperature Distribution on a Two-Dimensional Domain with Various Boundary Conditions

This example provides a line by line description of an ELLPACK program that will solve the problem posed in Section A.5 of Chapter II.

It is good practice to invoke the OPTIONS segment at the beginning of an ELLPACK program. In order to invoke the TIME and MEMORY modules write:

```
OPTIONS.          TIME $ MEMORY
```

The '$' symbol place between 'TIME' and 'MEMORY' is equivalent to combining two lines of programming. The keyword 'OPTIONS' which marks the segment block starts in column one and is followed by a period. The abbreviation OPT. is also recognized by

ELLPACK as the beginning of the OPTIONS segment. In general, ELLPACK will recognize the first three letters of any segment name.

Next the equation, boundary conditions, and grid are defined:

```
EQUATION.     UXX + UYY     =  -200.

BOUNDARY.     U             =  100.    ON X = 0.
              10*U + UX     =  1000.   ON X = 3.
              UY            =  0.      ON Y = 0.
                                       ON Y = 2.

GRID.         4 X POINTS
              3 Y POINTS
```

The spacing within each segment is established with ease of reading in mind and does not affect the execution of the ELLPACK program. In the BOUNDARY segment there is no <condition> statement preceding the code, 'ON Y = 2.'. This is a time saving feature. When a <condition> statement does not precede a <piece> statement in the BOUNDARY segment, ELLPACK uses the previous <condition> statement. In the GRID segment the <point list> statement is missing indicating that a uniform grid is superimposed onto the region.

ELLPACK now has the information necessary to invoke a discretization routine:

```
DISCRETIZATION.   5 POINT STAR
```

Since the problem is two dimensional, the 5 POINT module is used. For three dimensional problems, we use the 7 POINT STAR module.

At this point a solution method is specified:

SOLUTION. BAND GE

The BAND GE module requires a large amount of memory but works quite well as long as the matrix is small. For large matrices iterative methods are recommended.

All that remains is an output specification. For this problem, a table of data will be requested:

OUTPUT. TABLE(U)

Finally the END segment is invoked to mark the end of the ELLPACK program.

This short ELLPACK program accomplishes in 11 lines what an equivalent FORTRAN program would require hundreds of lines to accomplish. In reality the ELLPACK program is converted to a rather bulky FORTRAN program. However, the ease in which ELLPACK modules are invoked attest to ELLPACK's usefulness in solving elliptic equations. The output for this problem will verify the solutions obtained earlier in Chapter II.

2. Sample Program Two: Solving the Heat Equation on a Two Dimensional Chip Using a Non-uniform Grid

This example lists an ELLPACK program used for solving the non-uniform grid problem posed in Section B of Chapter II. The program is listed on the following page.

```
OPTIONS.        TIME $ MEMORY

EQUATION.         UXX + UYY          =  -50.

BOUNDARY.       A(Y)*U + B(Y)*UX   =  C(Y)  ON  X = 0.
                                            ON  X = 4.

GRID.           5 X POINTS
                7 Y POINTS  0., .5, 1., 1.5, 2.5, 3.5, 4.5

DISCRET.        5 POINT STAR

SOLUTION.       BAND GE

OUTPUT.         PLOT(U)

SUBPROGRAMS.

   REAL FUNCTION A(Y)
   A = 0.0
   IF (Y .GE. 1.5) A = 1.0
   RETURN
   END

   REAL FUNCTION B(Y)
   B = 1.0
   IF (Y .GE. 1.5) B = 0.0
   RETURN
   END

   REAL FUNCTION C(Y)
   C = 0.0
   IF (Y .GE. 1.5) C = 100.0
   RETURN
   END

END.
```

This example demonstrates the use of simple FORTRAN functions to model boundary conditions that are slightly more complex than those previously posed. Recall that, in the problem, on the x boundary there is a perfect insulator up to y = 1.5. The equation that governs the boundary condition on this segment of the x boundary is

$$\frac{\partial u}{\partial n} = 0,$$

where n is the outward pointing normal. Above y = 1.5 the boundary simply is u = 100. The <condition> statement for the x boundary in the BOUNDARY segment uses the FORTRAN functions A(Y) and B(Y) as variable coefficients and the FORTRAN function C(Y) as a variable right hand side to accurately depict both boundary conditions along x = 0 in a single boundary equation.

Additional features demonstrated in this example are the ELLPACK PLOT module invoked in the OUTPUT segment and the use of the SUBPROGRAM in which all FORTRAN functions and subroutines are placed.

F. LIMITATIONS OF ELLPACK

Two major limitations of the ELLPACK system preclude an attempt to solve the problem posed in Chapter I. First ELLPACK requires that a three dimensional domain be a box. Figure 1.4 depicts a chip mounted on a substrate in a configuration that is not a box. Current ELLPACK routines are unable to solve for this geometry. Secondly, ELLPACK routines do not accommodate a region in which two materials with different thermal properties come into physical contact. Chapters IV and V develop methods and

equations to be incorporated into FORTRAN and SUBPROGRAM segments. The methods are used to overcome the specified limitations and enable ELLPACK to solve the problem.

IV. EQUATIONS ALONG A CONVECTIVE INTERFACE

A. GEOMETRIC CONSIDERATIONS

To solve three dimensional elliptic problems using ELLPACK, the region under investigation must be a box. This restriction precludes the use of ELLPACK to solve the problem of Chapter I. As a technique to solve the problem we consider a region that is not a box, as illustrated in Figure 1.4, and enclose it in a box. The equations that govern the temperature profile for the subregions within the box can then be creatively constructed into a single ELLPACK expression over the box.

Figure 4.1 depicts this technique and establishes three subregions that then compose the focus of a solution. Subregion 1 is the chip which contains a heat source. Subregion 2 is the substrate which contains no heat source. Subregion 3 is an ambient temperature fluid region to which the heat generated in the chip is transferred via convection. The combined regions form a rectangular box as required by ELLPACK. Recall from Chapter I the variables which will be used throughout this discussion:

- k_i: Thermal conductivity coefficient for each subregion (note that k_3 for the ambient temperature fluid region is not defined).
- h_j: Convective coefficient along the subregion boundaries where convection is the mode of heat transfer.
- $\dot{q}_i$: Heat source for each subregion (the values of q_2 and q_3 for the substrate and the ambient temperature fluid, respectively, are equal to zero by the problem definition).

- u_∞: Ambient temperature of Subregion 3.
- T_1: The thickness of the chip.
- T_2: The thickness of the substrate (Note $T_2 << T_1$).
- L_1: The length of the chip.
- L_2: The width of the chip.
- L: The length and width of the substrate.

The following assumptions are made:

- The primary mode of heat transfer from Subregion 1 to Subregion 2 is conduction.
- The primary mode of heat transfer from Subregion 1 to Subregion 3 is convection.
- The primary mode of heat transfer from Subregion 2 to Subregion 3 is convection.
- The ambient temperature throughout the fluid in Subregion 3 is constant.
- The convection coefficient is constant for all convective heat transfer surfaces.

B. THE ELLPACK EQUATION SEGMENT FOR THE MODIFIED REGION

Three different elliptic equations model the temperature profile for the three subregions. Temperatures in the chip are governed by Poisson's equation

$$u_{xx} + u_{yy} + u_{zz} = \frac{-\dot{q}_1}{k_1}, \tag{4.1}$$

where $\dot{q}_1$ is the heat generated inside the chip. In the substrate the temperature profile is governed by Laplace's equation

$$u_{xx} + u_{yy} + u_{zz} = 0. \tag{4.2}$$

In the fluid region the temperature is known, e.g.

$$u = u_{\infty} . \tag{4.3}$$

Equations 4.1, 4.2, and 4.3 can be combined into a single general equation written in self-adjoint form as per ELLPACK requirements:

$$(A(x,y,z)u_x)_x + (A(x,y,z)u_y)_y + (A(x,y,z)u_z)_z + B(x,y,z)u = C(x,y,z), \tag{4.4}$$

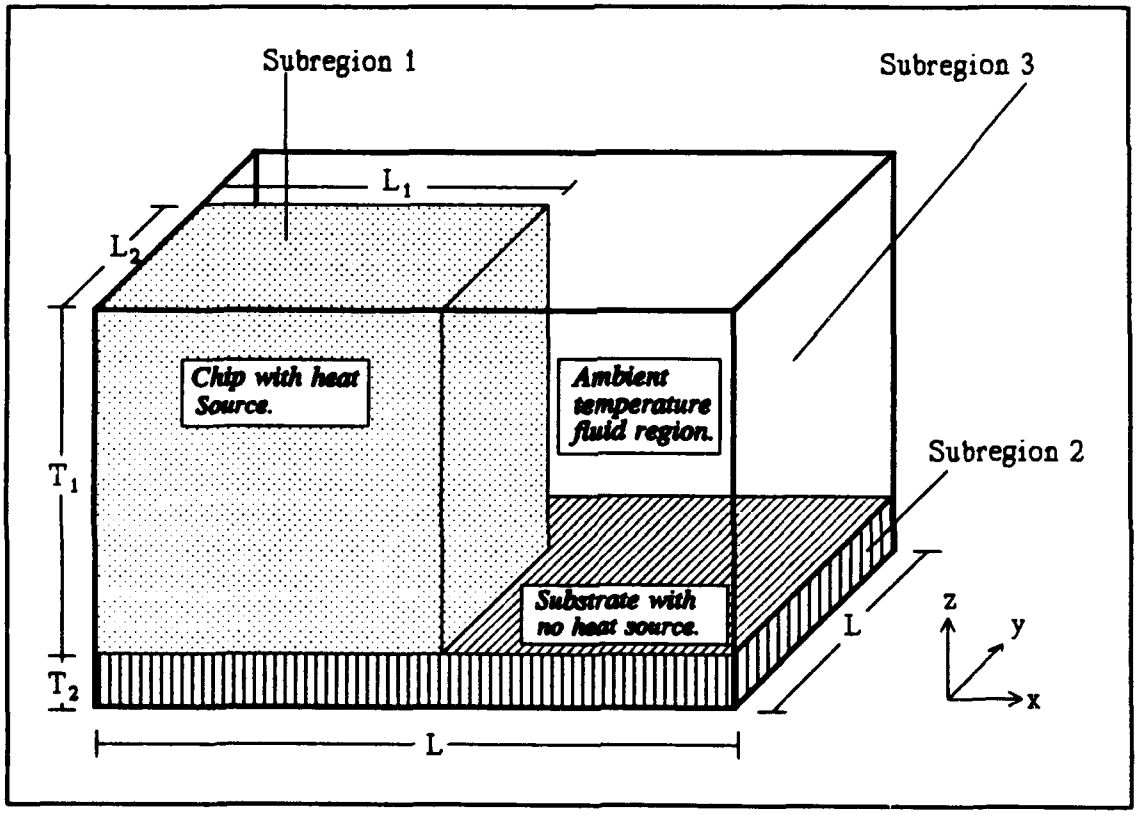


Figure 4.1: Three dimensional rectangular region subdivided into three subregions.

where the coefficients A(x,y,z), B(x,y,z), and C(x,y,z) are constants within their respective subregions. Table 4.1 is a summary for the coefficient values in the three subregions.

TABLE 4.1: SUMMARY OF EQUATION COEFFICIENT VALUES

	Subregion		
Coefficient	1	2	3
A(x,y,z)	1	1	0
B(x,y,z)	0	0	1
C(x,y,z)	$-\dot{q}_1/k_1$	0	u_∞

These variable coefficient expressions can be defined as FORTRAN functions that assign appropriate values based on the subregion in which the node lies. The FORTRAN functions are included in the SUBPROGRAM segment of the ELLPACK program. A FORTRAN function that would describe the coefficient C(x,y,z) can be written as:

```
REAL FUNCTION C(X,Y,Z)
REAL K1, L1, L2
COMMON T2, K1, L1, L2, Q1, UINF
C = 0.0
IF (Z .LT. T2) RETURN
C = UINF
IF (X .LE. L1/2. .AND. Y .LE. L2/2) C = -Q1/K1
RETURN
END
```

C. THE ELLPACK BOUNDARY SEGMENT FOR THE MODIFIED REGION

The ELLPACK BOUNDARY segment for the modified region is set up using the same techniques used in previous section to handle the EQUATION segment. There are six boundaries to consider. These are depicted in Figure 4.2. For example, for the upper boundary on the top of the box in the area occupied by the chip, heat convects to the surrounding fluid medium. Recall that convection is modeled by Newton's Law of Cooling which, on the upper boundary of a region, generates

$$hu + ku_z = hu_\infty \ . \tag{4.5}$$

On the top of the box, in the area not occupied by the chip, the boundary is governed by the Dirichlet condition

$$u = u_\infty \ . \tag{4.6}$$

Equations 4.5 and 4.6 can be combined in the following general form

$$FU1(x,y)u_z + FU2(x,y)u = FU2(x,y)u_\infty \ , \tag{4.7}$$

where FU1(x,y) and FU2(x,y) are functions that establish the value of the coefficients based on the location of the node within the upper boundary. As in the EQUATION segment, these expressions are written as FORTRAN functions and inserted in the SUBPROGRAM segment of the ELLPACK program.

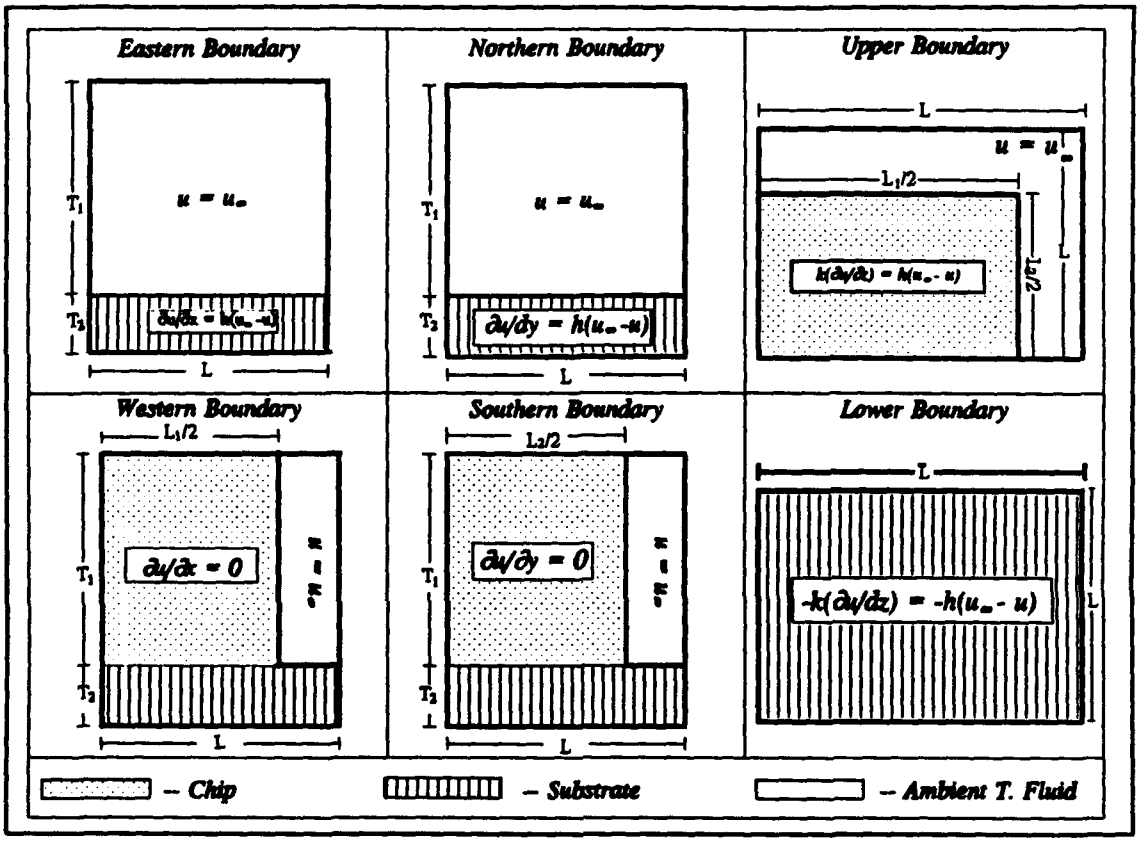


Figure 4.2: Boundary conditions for region depicted in Figure 4.1.

D. EQUATIONS ALONG THE CONVECTIVE INTERFACE

After the EQUATION and BOUNDARY segments are correctly coded the ELLPACK program is executed. ELLPACK discretizes the problem, as defined in the box, and solves for the resulting linear system. Unfortunately, the solution to this system is an incorrect representation of the temperature profile for the region. The reason for the erroneous results is that the convective heat transfer has not been modeled for the nodes that lie on the interfaces internal to the box where the chip and substrate come in contact with each other and with the fluid region. The internal convective boundary between the fluid region and solid regions is referred to as a convective interface. If the fluid region

lies to the east of the chip or substrate, then the interface will be referred to as an eastern convective interface, and so forth.

The solution can be corrected by replacing the ELLPACK generated equations with equations that model the convection process for the nodes that lie along the convective interface. Recall that the convective boundary condition is modeled by Newton's Law of Cooling. On an east or west facing convective boundary Newton's Law of Cooling is written as

$$k\frac{\partial u}{\partial n} = h(u_\infty - u), \tag{4.8}$$

where n is the outward pointing normal along the boundary. Recall further that the outward pointing normal derivative is not the same as the regular derivative on a given boundary or interface. On a western boundary the outward pointing normal derivative points in the negative x direction. Therefore, on a western boundary

$$\frac{\partial u}{\partial n} = -\frac{\partial u}{\partial x}. \tag{4.9}$$

Using the regular derivative, Newton's Law of Cooling on the western convective boundary becomes

$$-k\frac{\partial u}{\partial x} = h(u_\infty - u). \tag{4.10}$$

Equation 4.10 can be rewritten as

$$k\frac{\partial u}{\partial x} - hu = -hu_{\infty}, \tag{4.11}$$

which then appears in ELLPACK format as

$$K*UX - H*U = -H*UINF. \tag{4.12}$$

The considerations discussed here must also be applied at the southern and lower boundaries. For eastern, northern, and upper boundaries the derivative in the direction of the outward pointing normal is the same as the regular derivative. Newton's Law of Cooling at an eastern convective boundary is

$$k\frac{\partial u}{\partial x} = h(u_{\infty} - u), \tag{4.13}$$

which then is rewritten as

$$k\frac{\partial u}{\partial x} + hu = hu_{\infty}, \tag{4.14}$$

and translated into an ELLPACK expression as

$$K*UX + H*U = T*UINF. \tag{4.15}$$

It is now possible to replace the linear equations for all nodes that lie on eastern and western convective interfaces with equations that model the convection.

Consider an eastern convective interface. Recall from Chapter II that for a nodal point lying on a Neumann or mixed boundary, the grid is uniform in the direction normal to the interface. This is true because a fictitious point was introduced when the seven point nodal equation for a non-uniform grid was invoked. The fictitious point was chosen so that it was the same distance from the interface as its non-fictitious counterpart. Thus $\alpha = 1$ on the boundary node. Substituting this into Equation 2.39 generates the equation

$$\begin{aligned} &\frac{1}{2(\Delta x)^2}u_E + \frac{1}{\beta(\beta+1)(\Delta y)^2}u_N + \frac{1}{\gamma(\gamma+1)(\Delta z)^2}u_U \\ &\quad - \left(\frac{1}{(\Delta x)^2} + \frac{1}{\beta(\Delta y)^2} + \frac{1}{\gamma(\Delta z)^2}\right)u_C + \frac{1}{2(\Delta x)^2}u_W \\ &\quad + \frac{1}{(\beta+1)(\Delta y)^2}u_S + \frac{1}{(\gamma+1)(\Delta z)^2}u_L = -\frac{\dot{q}}{2k}. \end{aligned} \tag{4.16}$$

On the eastern convective interface the nodal point u_E is fictitious and must be eliminated using the convective boundary condition given by Equation 4.13. Using the central difference approximations of the first derivative on a uniform grid, Equation 4.13 is rewritten as

$$k\left(\frac{u_E - u_W}{2\Delta x}\right) = h(u_\infty - u_C), \tag{4.17}$$

which when solved for u_E becomes

$$u_E = \left(\frac{2h\Delta x}{k}\right)(u_\infty - u_C) + u_W . \tag{4.18}$$

Substituting Equation 4.18 in Equation 4.16 yields

$$\begin{aligned} &\frac{1}{\beta(\beta+1)(\Delta y)^2}u_N + \frac{1}{\gamma(\gamma+1)(\Delta z)^2}u_U \\ &- \left(\frac{1}{(\Delta x)^2}\left(1+\frac{h}{k}\Delta x\right) + \frac{1}{\beta(\Delta y)^2} + \frac{1}{\gamma(\Delta z)^2}\right)u_C + \frac{1}{(\Delta x)^2}u_W \\ &+ \frac{1}{(\beta+1)(\Delta y)^2}u_S + \frac{1}{(\gamma+1)(\Delta z)^2}u_L = -\frac{1}{k}\left(\frac{\dot{q}}{2} + \frac{hu_\infty}{\Delta x}\right)u. \end{aligned} \tag{4.19}$$

Equation 4.19 is used to replace the equations that were originally set by ELLPACK for nodes falling in an eastern convective interface.

To write a similar equation on a western convective interface, Equation 4.10 is rewritten using the central difference formula for the first derivative

$$-k\left(\frac{u_E - u_W}{2\Delta x}\right) = h(u_\infty - u_C), \tag{4.20}$$

which when solved for u_W becomes

$$u_W = \left(\frac{2h\Delta x}{k}\right)(u_\infty - u_C) + u_E \ . \tag{4.21}$$

Note that exchanging u_E and u_W in Equation 4.21 gives the same results that were obtained in Equation 4.18. Substituting Equation 4.21 into Equation 4.19 gives

$$\begin{aligned} &\frac{1}{(\Delta x)^2}u_E + \frac{1}{\beta(\beta+1)(\Delta y)^2}u_N + \frac{1}{\gamma(\gamma+1)(\Delta z)^2}u_U \\ &- \left(\frac{1}{(\Delta x)^2}\left(1+\frac{h}{k}\Delta x\right) + \frac{1}{\beta(\Delta y)^2} + \frac{1}{\gamma(\Delta z)^2}\right)u_C \\ &+ \frac{1}{(\beta+1)(\Delta y)^2}u_S + \frac{1}{(\gamma+1)(\Delta z)^2}u_L = -\frac{1}{k}\left(\frac{\dot{q}}{2} + \frac{hu_\infty}{\Delta x}\right). \end{aligned} \tag{4.22}$$

It is not surprising that Equation 4.22 is the same as Equation 4.19 when u_W is substituted for u_E.

A similar derivation for a northern convective interface generates

$$\begin{aligned} &\frac{1}{\alpha(\alpha+1)(\Delta x)^2}u_E + \frac{1}{\gamma(\gamma+1)(\Delta z)^2}u_U \\ &- \left(\frac{1}{\alpha(\Delta x)^2} + \frac{1}{(\Delta y)^2}\left(1+\frac{h}{k}\Delta y\right) + \frac{1}{\gamma(\Delta z)^2}\right)u_C + \frac{1}{(\Delta y)^2}u_S \\ &+ \frac{1}{(\alpha+1)(\Delta x)^2}u_W + \frac{1}{(\gamma+1)(\Delta z)^2}u_L = -\frac{1}{k}\left(\frac{\dot{q}}{2} + \frac{hu_\infty}{\Delta y}\right). \end{aligned} \tag{4.23}$$

The equation for the southern convective interface is obtained by substituting u_N for u_S in Equation 4.23.

Finally on an upper convective interface the nodal equation becomes

$$\frac{1}{\alpha(\alpha+1)(\Delta x)^2}u_E + \frac{1}{\beta(\beta+1)(\Delta y)^2}u_N - \left(\frac{1}{\alpha(\Delta x)^2} + \frac{1}{\beta(\Delta y)^2} + \frac{1}{(\Delta z)^2}\left(1+\frac{h}{k}\Delta z\right)\right)u_C + \frac{1}{(\Delta z)^2}u_L + \frac{1}{(\alpha+1)(\Delta x)^2}u_W + \frac{1}{(\beta+1)(\Delta y)^2}u_S = -\frac{1}{k}\left(\frac{\dot{q}}{2} + \frac{hu_\infty}{\Delta z}\right). \tag{4.24}$$

The equation on a lower convective interface is generated by substituting u_U for u_L in Equation 4.24.

E. EQUATIONS AT NODES WHERE TWO OR MORE CONVECTIVE INTERFACES INTERSECT.

Consider the hypothetical region in Figure 4.3 which is a cross sectional slice along the z axis of a three dimensional region. Subregion 1 is a solid material with a heat source. Subregion 2 is an ambient temperature fluid region. The mechanism for heat transfer along the interface is convection. ELLPACK generates the correct equations for Nodes 1 and 9 since these nodes fall on the boundary. The equations at Nodes 2 and 6 must be replaced by Equation 4.9 which models the eastern convective interface. The

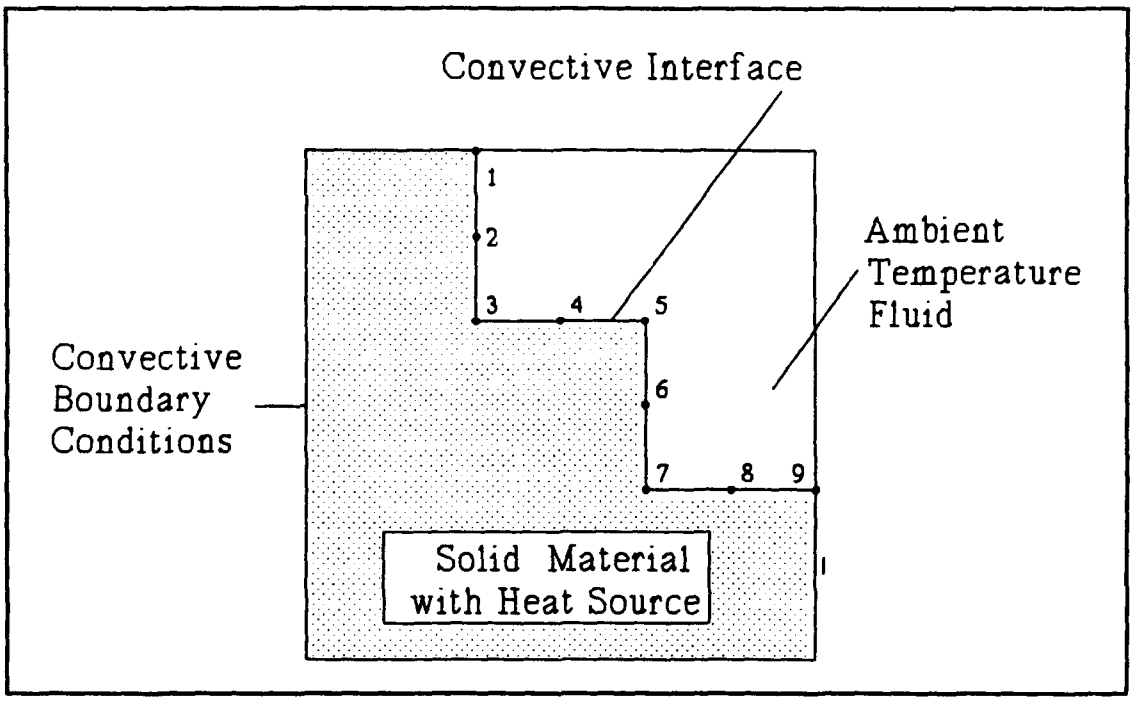


Figure 4.3: Two dimensional region depicting intersections between convective interfaces.

equations for Nodes 4 and 8 must be replaced with Equation 4.23 which models the northern convective interface.

The remaining nodes lie at the intersection of two convective interfaces. At Node 5 the intersection is convex in nature with respect to Subregion 1. At Nodes 3 and 7 the intersection is concave in nature with respect to Subregion 2. These basic intersections must be treated differently.

When setting up the seven point nodal equation at Node 5, two fictitious points, u_E and u_N, are introduced in the expression. The fictitious nodes are substituted with expressions derived from Newton's Law of Cooling. The expression for u_E is shown in Equation 4.18. By a similar derivation the expression for u_N is

$$u_N = \left(\frac{2h\Delta x}{k}\right)(u_\infty - u_C) + u_S \,. \tag{4.25}$$

At the interface, the grid is made uniform in both the x and y directions when accommodating the fictitious nodes, hence $\alpha = 1$ and $\beta = 1$. Substituting this into Equation 2.39 generates

$$\begin{aligned} &\frac{1}{2(\Delta x)^2}u_E + \frac{1}{2(\Delta y)^2}u_N + \frac{1}{\gamma(\gamma+1)(\Delta z)^2}u_U \\ &\quad - \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{\gamma(\Delta z)^2}\right)u_C + \frac{1}{2(\Delta x)^2}u_W \\ &\quad + \frac{1}{2(\Delta y)^2}u_S + \frac{1}{(\gamma+1)(\Delta z)^2}u_L = -\frac{\dot{q}}{2k}. \end{aligned} \tag{4.26}$$

Substituting the appropriate expressions for u_E and u_N gives

$$\begin{aligned} &\frac{1}{\gamma(\gamma+1)(\Delta z)^2}u_U - \left(\frac{1}{(\Delta x)^2}\left(1 + \frac{h}{k}\Delta x\right) + \frac{1}{(\Delta y)^2}\left(1 + \frac{h}{k}\Delta y\right) + \frac{1}{\gamma(\Delta z)^2}\right)u_C \\ &\quad + \frac{1}{(\Delta x)^2}u_W + \frac{1}{(\Delta y)^2}u_S + \frac{1}{(\gamma+1)(\Delta z)^2}u_L = -\frac{1}{k}\left(\frac{\dot{q}}{2} + hu_\infty\left(\frac{1}{\Delta x} + \frac{1}{\Delta y}\right)\right). \end{aligned} \tag{4.27}$$

Equation 4.27 is used to replace the ELLPACK generated equation at Node 5. Similar modifications can be made along any intersection of convex type between convective interfaces.

When setting up the seven point nodal equation at Nodes 3 and 7, no fictitious points are introduced into the expression. This suggests that no modifications need be made to the ELLPACK generated equations for these nodes. However, these nodes are affected by convection. Consider the equation at Node 3 which would include the terms u_2 and u_4 from its neighboring nodes to the east and north. The equations at Nodes 2 and 4 have been modified to model the convective process by replacing ELLPACK generated equations with Equations 4.19 and 4.23, respectively. Since the terms u_2 and u_4 appear in the difference equation at Node 3, convection is modeled implicitly here. Convection is also modeled implicitly at Node 7.

F. RUNNING THE PROGRAM WITH CONVECTIVE INTERFACE MODIFICATIONS

The actual programming mechanics involved in changing the ELLPACK generated equations for nodes lying on convective interfaces are discussed in Chapter VI. Once these modifications have been made, the ELLPACK program can be successfully executed. However, application of ELLPACK to the problem of Chapter I requires further manipulation of the ELLPACK generated equations for nodes lying on the interface between the substrate and the chip. The primary heat transfer mechanism at this interface is conduction. The modifications required to mathematically model this interface are discussed in Chapter V.

V. EQUATIONS ALONG A CONDUCTIVE INTERFACE

The equations derived in this chapter are used to replace ELLPACK generated equations for nodes that lie along the interface between the chip and the substrate. In general, these equations model conductive heat transport between two solid materials that have different thermal properties. The interface between two materials in which conduction is the mode of heat transfer is called the conductive interface.

A. ONE DIMENSIONAL ANALYSIS

1. Equation Derivation

Consider the one dimensional region depicted in Figure 5.1. Two materials with different thermal properties come into thermal contact at point $x = x_n$. The following variables are used to describe these thermal properties:

- k_A: Thermal conductivity in Material A.
- k_B: Thermal conductivity in Material B.
- $\dot{q}_A$: Heat source in Material A.
- $\dot{q}_B$: Heat source in Material B.

A non-uniform grid is superimposed onto the region. In Figure 5.1, u_i and v_i denote the temperatures at node i in Material A and Material B respectively. The following variables are defined for the nodal network in Figure 5.1:

$$
\begin{aligned}
u_E^* &= u(x_{n+1}), & v_E &= v(x_{n+1}), \\
u_C &= u(x_n), & v_C &= v(x_n), \\
u_W &= u(x_{n-1}), & v_W^* &= v(x_{n-1}),
\end{aligned}
\tag{5.1}
$$

where u(x) and v(x) are functions that describe the temperature profile in Materials A and B respectively. The nodal network on the conductive interface introduces two fictitious points, u_E^* and v_W^*, which have been flagged with asterisks so that they can be easily tracked in the derivation that follows. These terms appear in the numerical equations developed along the conductive interface but are eliminated in the final form of the equation.

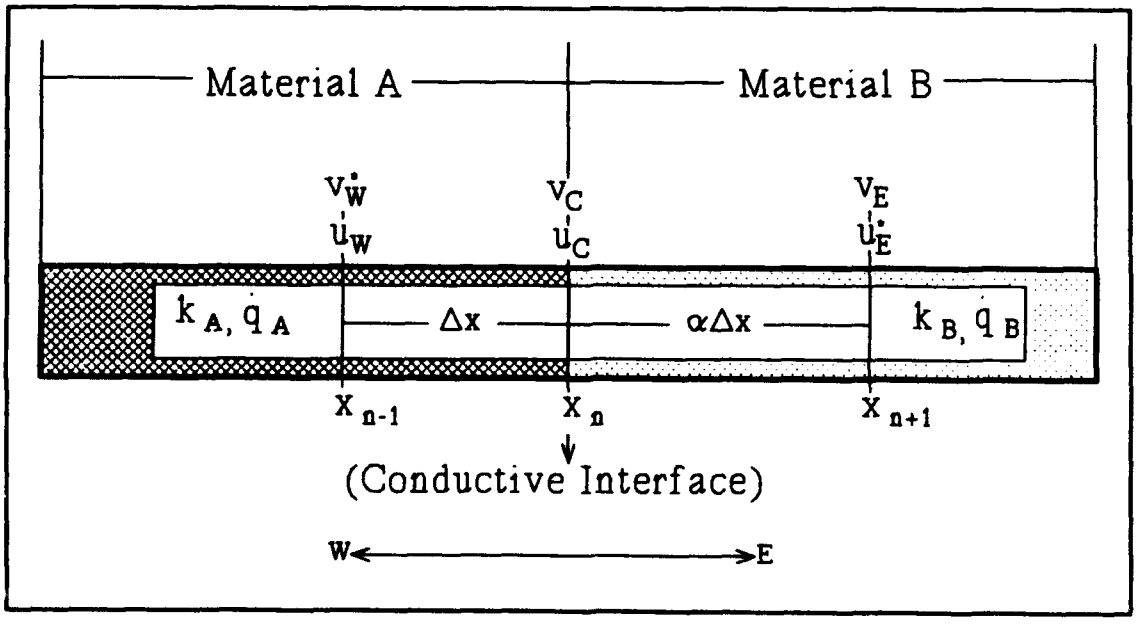


Figure 5.1: Non-uniform grid superimposed onto a one dimensional region consisting of two materials that come into thermal contact along a conductive interface.

Heat transport along the conductive interface is modeled by Poisson's equation. Equation 2.39, modified for one dimensional analysis, is used to approximate Poisson's equation along the conductive interface. In terms of both u and v the equations become

$$u_E^* - (\alpha + 1)u_c + \alpha u_W = \frac{-\alpha(\alpha + 1)(\Delta x)^2 \dot{q}_A}{2k_A} , \tag{5.2a}$$

$$v_E - (\alpha + 1)v_c + \alpha v_W^* = \frac{-\alpha(\alpha + 1)(\Delta x)^2 x \dot{q}_b}{2k_b} . \tag{5.2b}$$

The fictitious variables u_E^* and v_W^* are eliminated by imposing conditions of continuity at the conductive interface. The first continuity condition states that the temperature of Material A and the temperature of Material B must be equal at the conductive interface. Hence,

$$u_C = v_C . \tag{5.3}$$

The second continuity condition states that the heat flux through the eastern boundary of Material A must be the same as the heat flux through the western boundary of Material B. Therefore, at the conductive interface,

$$k_A \frac{du}{dx}\Big|_C = k_B \frac{dv}{dx}\Big|_C . \tag{5.4}$$

Substituting Equation 2.34 into Equation 5.4 yields

$$k_A\left(\frac{u_E^{\bullet} + (\alpha^2 - 1)u_C - \alpha^2 u_W}{\alpha(\alpha + 1)}\right) = k_B\left(\frac{v_E + (\alpha^2 - 1)v_C - \alpha^2 v_W^{\bullet}}{\alpha(\alpha + 1)}\right). \tag{5.5}$$

Solving Equation 5.5 for $u_E^{\bullet}$ gives

$$u_E^{\bullet} = Kv_E + K(\alpha^2 - 1)v_C - K\alpha^2 v_W^{\bullet} - (\alpha^2 - 1)u_C + \alpha^2 u_W\ , \tag{5.6}$$

where

$$K = \frac{k_A}{k_B}\ . \tag{5.7}$$

Substituting Equation 5.6 into Equation 5.2a yields

$$\begin{aligned} &Kv_E - (\alpha^2 + \alpha)u_C + K(\alpha^2 - 1)v_C + (\alpha^2 + \alpha)u_W \\ &\qquad - K\alpha^2 v_W^{\bullet} = -\frac{\alpha(\alpha + 1)(\Delta x)^2 \dot{q}_A}{2k_A}\ . \end{aligned} \tag{5.8}$$

Using the temperature continuity condition at the conductive interface, Equation 5.8 is rewritten as

$$\begin{aligned} &Kv_E - (-K\alpha^2 + \alpha^2 + K + \alpha)u_C + \alpha(\alpha + 1)u_W \\ &\qquad - K\alpha^2 v_W^{\bullet} = -\frac{\alpha(\alpha + 1)(\Delta x)^2 \dot{q}_A}{2k_A}\ . \end{aligned} \tag{5.9}$$

Now multiplying Equation 5.2b by Kα gives

$$K\alpha v_E - (K\alpha^2 + K\alpha)v_c + K\alpha^2 v_W^* = -\frac{\alpha^2(\alpha+1)(\Delta x)^2 \dot{q}_B}{2k_B} . \qquad (5.10)$$

The term v_W^* is eliminated by adding Equations 5.9 and 5.10. After applying the temperature continuity condition and simplifying the results are

$$\frac{K}{\alpha}v_E - \frac{2(K+\alpha)}{\alpha}u_C + u_W = -\frac{(\Delta x)^2}{2k_A}(\dot{q}_A + \alpha\dot{q}_B) . \qquad (5.11)$$

Equation 5.11 is used to generate equations for nodes on the conductive interface of a one dimensional region.

2. Example: Solving for the Temperature Profile in a One Dimensional Region with a Conductive Interface for which the Grid is Non-uniform

a. Problem Statement

This problem solves the temperature profile in a rod of length ten centimeters which consists of two different homogeneous materials. The materials form a conductive interface on the center of the rod.

b. Data

Data for the problem is summarized in Figure 5.2.

c. Analysis

The following assumptions apply in the analysis of this problem:

- No heat is lost through the side of the rod (e.g. the temperature is not a function of radius).
- The properties for Material A and Material B are constant throughout their respective regions.

The temperature at each node is denoted by the variable u_i. The grid interior to Material A is uniform, hence the approximation of the heat equation for these nodes is given by Equation 2.6. A Dirichlet boundary condition is incorporated in the equation for Node 1. The resulting equations for the nodes interior to Material A are

$$\begin{aligned} -2u_1 + u_2 &= 0, \\ u_1 - 2u_2 + u_3 &= 0, \\ u_2 - 2u_3 + u_4 &= 0, \\ u_3 - 2u_4 + u_5 &= 0. \end{aligned} \qquad (5.12a)$$

The conductive interface lies on Node 5. The heat equation here is approximated by Equation 5.11

$$2u_4 - 42u_5 + 40u_6 = -50. \qquad (5.12b)$$

The nodes interior to Material B are more densely packed than those in Material A. However, the grid is still uniform and Equation 2.6 is used to write the difference

equations at these nodes. The Dirichlet boundary condition is incorporated in the equation for Node 14. The resulting equations for the nodes interior to Material B are

$$
\begin{aligned}
u_5 - 2u_6 + u_7 &= -2.5\,, \\
u_6 - 2u_7 + u_8 &= -2.5\,, \\
\vdots \quad\quad \vdots \quad\quad \vdots \quad\quad &\quad\vdots \qquad\qquad (5.12c) \\
u_{12} - 2u_{13} + u_{14} &= -2.5\,, \\
u_{13} - 2u_{14} &= -102.5\,.
\end{aligned}
$$

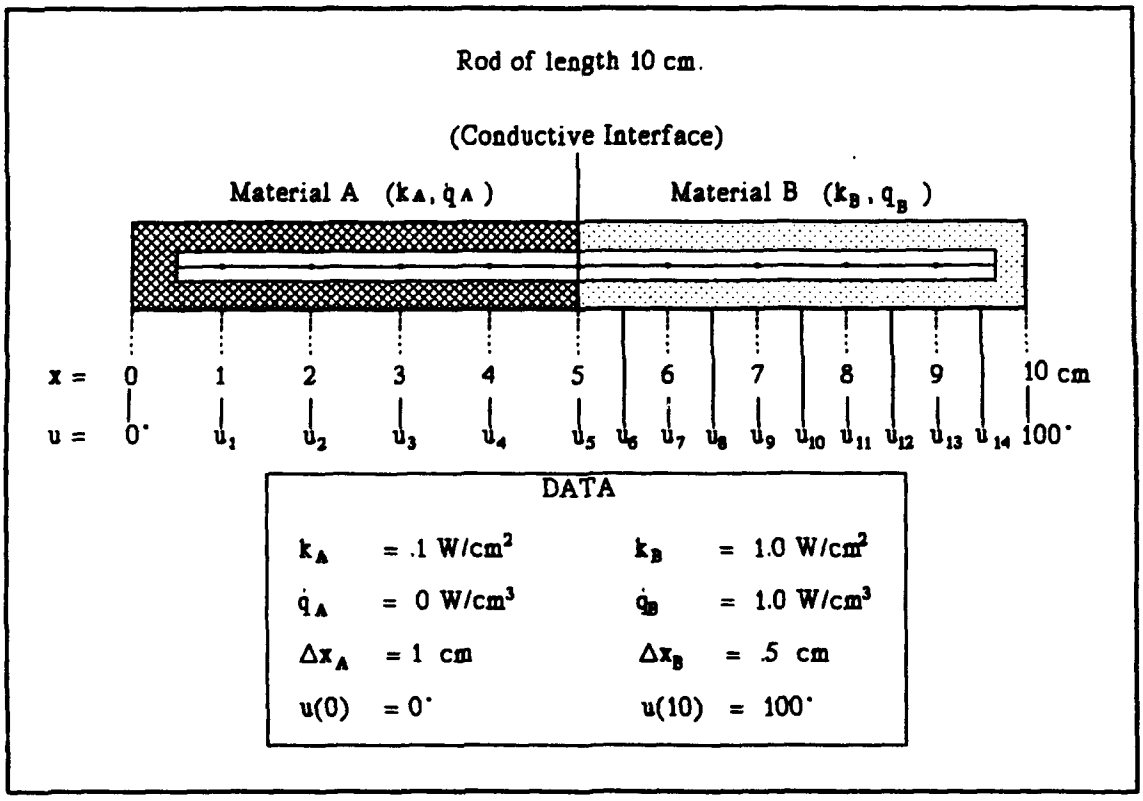


Figure 5.2: Data for one dimensional region in which two materials with different thermal properties come into contact along a conductive interface.

In matrix form, Equation 5.12 becomes

$$\begin{bmatrix} -2 & 1 & & & & & & & & & & & & \\ 1 & -2 & 1 & & & & & & & & & & & \\ & 1 & -2 & 1 & & & & & & & & & & \\ & & 1 & -2 & 1 & & & & & & & & & \\ & & & 1 & -42 & 40 & & & & & & & & \\ & & & & 1 & -2 & 1 & & & & & & & \\ & & & & & 1 & -2 & 1 & & & & & & \\ & & & & & & 1 & -2 & 1 & & & & & \\ & & & & & & & 1 & -2 & 1 & & & & \\ & & & & & & & & 1 & -2 & 1 & & & \\ & & & & & & & & & 1 & -2 & 1 & & \\ & & & & & & & & & & 1 & -2 & 1 & \\ & & & & & & & & & & & 1 & -2 & \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -50 \\ -2.5 \\ -2.5 \\ -2.5 \\ -2.5 \\ -2.5 \\ -2.5 \\ -2.5 \\ -2.5 \\ -102.5 \end{bmatrix} \quad (5.13)$$

d. Solution

The solution to Equation 5.13 is an approximation of the temperature at the nodes shown in Figure 5.2. The data is plotted in Figure 5.3. Since this problem is relatively simple it can be solved analytically. The result is:

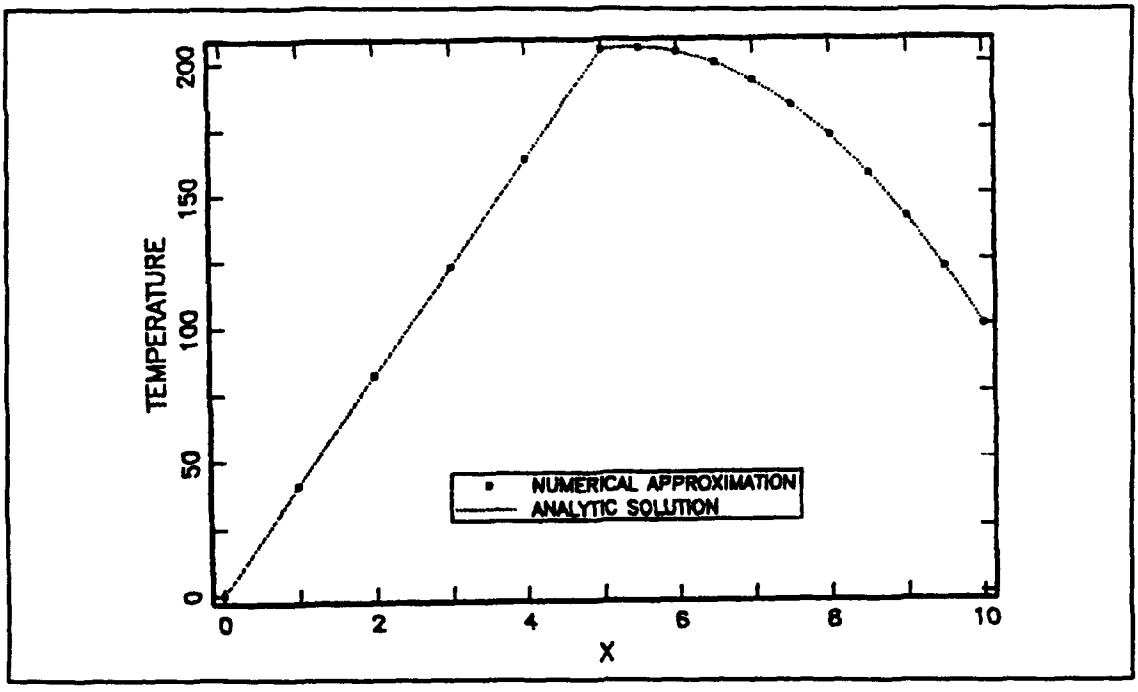


Figure 5.3: Analytical and numerical solutions to one dimensional conductive interface example.

$$u(x) = \left(\frac{450}{11}\right)x \qquad 0 \le x \le 5$$

$$v(x) = -5x^2 + \left(\frac{595}{11}\right)x + \frac{650}{11} \qquad 5 \le x \le 10. \tag{5.14}$$

Equation 5.14 is superimposed onto the numerical approximation in Figure 5.3. The fit is nearly perfect. This is not surprising since the temperature profile in Material A is shown to be linear. The temperature profile in Material B is quadratic and therefore has a constant second derivative. Both behaviors are modeled quite well by the second order difference approximations which were used to numerically obtain the temperature profile.

B. TWO DIMENSIONAL ANALYSIS

Development of the two dimensional analog of Equation 5.11 follows a similar methodology as that of the one dimensional counterpart. Figure 5.4 represents a section of a two dimensional region at a conductive interface that is oriented parallel to the y axis at $x = x_i$. A non-uniform grid is superimposed onto the region. As before, the variables u and v are used to describe the temperature profiles in Material A and Material B respectively. The following subscripted variables are defined:

$$\begin{aligned}
u_C &= u(x_i,y_j), & v_C &= v(x_i,y_j),\\
u_E^{\bullet} &= u(x_{i+1},y_j), & v_E &= v(x_{i+1},y_j),\\
u_W &= u(x_{i-1},y_j), & v_W^{\bullet} &= v(x_{i-1},y_j),\\
u_N &= u(x_i,y_{j+1}), & v_N &= v(x_i,y_{j+1}),\\
u_S &= u(x_i,y_{j-1}), & v_S &= v(x_i,y_{j-1}),
\end{aligned} \tag{5.15}$$

where u(x,y) and v(x,y) are functions that describe the temperature profile in Materials A and Materials B respectively. The variables $u_E^{\bullet}$ and $v_W^{\bullet}$ are fictitious and are eventually eliminated.

The approximation for Poisson's equation at the conductive interface comes from the two dimensional form of Equation 2.38. This equation approximates the heat flow in the region is expressed in terms of both u and v at the interface.

$$\frac{u_E^{\bullet} - (\alpha+1)u_C + \alpha u_W}{\alpha(\alpha+1)(\Delta x)^2} + \frac{u_N - (\beta+1)u_C + \beta u_S}{\beta(\beta+1)(\Delta y)^2} = -\frac{\dot{q}_A}{2k_A}, \tag{5.16a}$$

$$\frac{v_E - (\alpha+1)v_C + \alpha v_W^*}{\alpha(\alpha+1)(\Delta x)^2} + \frac{v_N - (\beta+1)v_C + \beta v_S}{\beta(\beta+1)(\Delta y)^2} = -\frac{\dot{q}_B}{2k_B} \quad . \tag{5.16b}$$

As in the one dimensional analysis, the following conditions of continuity are imposed at the interface:

$$\begin{gathered} u_N = v_N, \quad u_C = v_C, \quad u_S = v_S, \\ k_A \frac{\partial u}{\partial x}\Big|_C = k_B \frac{\partial v}{\partial x}\Big|_C \quad . \end{gathered} \tag{5.17}$$

In order to simplify the analysis, N is denoted as

$$N = \frac{u_N - (\beta+1)u_C + \beta u_S}{\beta(\beta+1)(\Delta y)^2} \quad . \tag{5.18a}$$

By the continuity conditions in Equation 5.17, N can also be expressed as

$$N = \frac{v_N - (\beta+1)v_C + \beta v_S}{\beta(\beta+1)(\Delta y)^2} \quad . \tag{5.18b}$$

Substituting N into Equation 5.16 yields

$$u_E^* - (\alpha+1)u_C + \alpha u_W = -\alpha(\alpha+1)(\Delta x)^2 \left(N + \frac{\dot{q}_A}{2k_A} \right), \tag{5.19a}$$

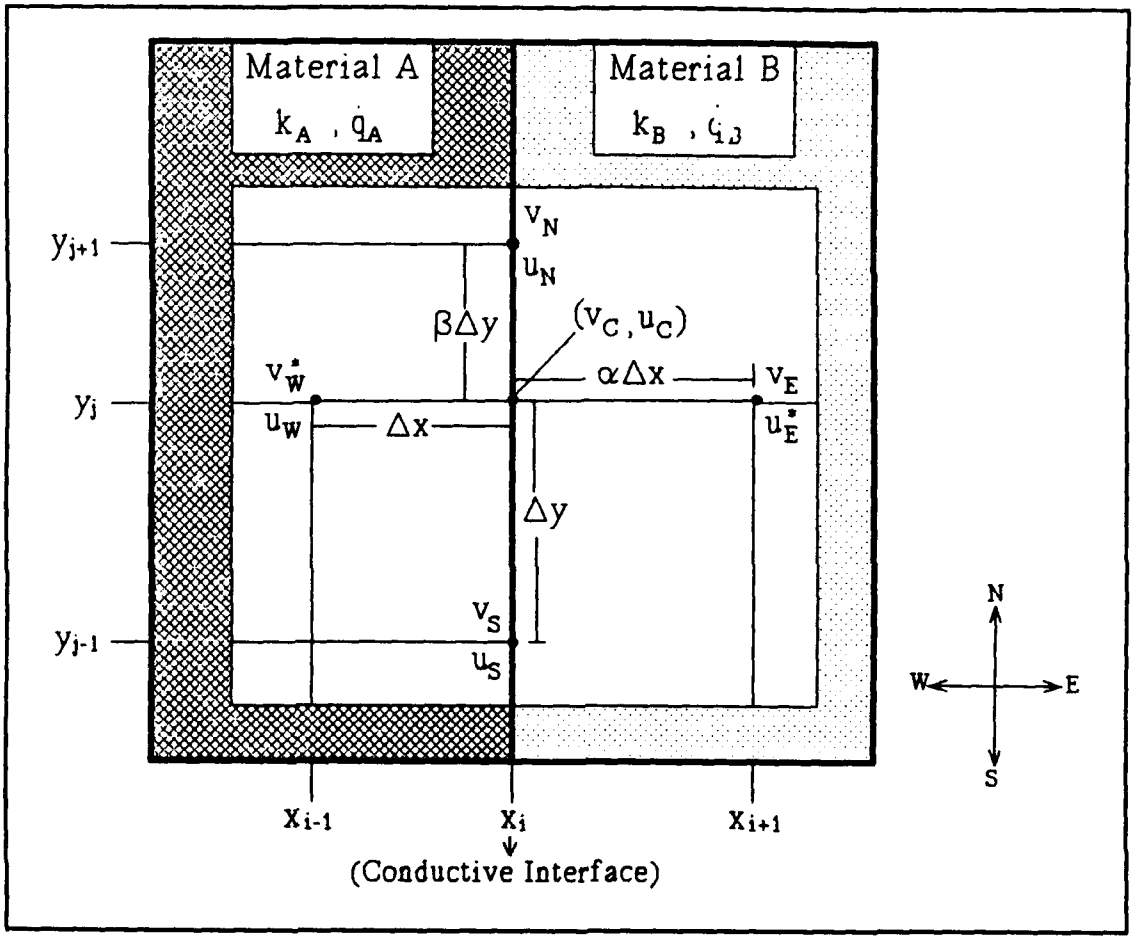


Figure 5.4: Non-uniform grid superimposed onto a two dimensional region in which two materials come into thermal contact along a conductive interface.

$$v_E - (\alpha + 1)v_C + \alpha v_W^* = -\alpha(\alpha + 1)(\Delta x)^2 \left(N + \frac{\dot{q}_B}{2k_B} \right). \qquad (5.19b)$$

If the proper substitution is made for the right hand side, Equation 5.19 is the same as Equation 5.2. Recognizing this fact enables one to use Equation 5.19 to perform the same derivation that leads to the conductive interface equation in one dimension. The result in two dimensions is

$$\frac{K}{\alpha}v_E - \frac{(K+\alpha)}{\alpha}u_C + u_W = -(\Delta x)^2\left(\frac{\dot{q}_A + \alpha\dot{q}_B}{2k_A} + (\alpha+1)N\right). \tag{5.20}$$

Now substituting Equation 5.18a into Equation 5.20 generates

$$\frac{K}{\alpha}v_E + \frac{(\alpha+1)(\Delta x)^2}{\beta(\beta+1)(\Delta y)^2}u_N + \left(\frac{K+\alpha}{\alpha} + \frac{(\alpha+1)(\Delta x)^2}{\beta(\Delta y)^2}\right)u_C$$
$$+ \frac{(\alpha+1)(\Delta x)^2}{(\beta+1)(\Delta y)^2}u_S + u_W = -\frac{(\Delta x)^2}{2k_A}(\dot{q}_A + \alpha\dot{q}_B)\,. \tag{5.21}$$

Equation 5.21 correctly approximates heat transport on a conductive interface that is parallel to the y axis. If the conductive interface is parallel to the x axis a similar analysis results in

$$\frac{K}{\beta}v_N + \frac{(\beta+1)(\Delta y)^2}{\alpha(\alpha+1)(\Delta x)^2}u_E + \left(\frac{K+\beta}{\beta} + \frac{(\beta+1)(\Delta y)^2}{\alpha(\Delta x)^2}\right)u_C$$
$$+ \frac{(\beta+1)(\Delta y)^2}{(\alpha+1)(\Delta x)^2}u_W + u_S = -\frac{(\Delta y)^2}{2k_A}(\dot{q}_A + \beta\dot{q}_B)\,. \tag{5.22}$$

C. THREE DIMENSIONAL ANALYSIS

1. Equation Derivation

The derivation of the three dimensional form of the heat equation at the conductive interface of a two material region is the same as that of its lower dimensional forms. If the conductive interface is parallel to the yz plane, the equation centered at node u_C is

$$\frac{K}{\alpha}v_E + \frac{(\alpha+1)(\Delta x)^2}{\beta(\beta+1)(\Delta y)^2}u_N + \frac{(\alpha+1)(\Delta x)^2}{\gamma(\gamma+1)(\Delta z)^2}u_U$$

$$-\left(\frac{K+\alpha}{\alpha} + \frac{(\alpha+1)(\Delta x)^2}{\beta(\Delta y)^2} + \frac{(\alpha+1)(\Delta x)^2}{\gamma(\Delta z)^2}\right)u_C \qquad (5.23)$$

$$+\frac{(\alpha+1)(\Delta x)^2}{(\gamma+1)(\Delta z)^2}u_L + \frac{(\alpha+1)(\Delta x)^2}{(\beta+1)(\Delta y)^2}u_S + u_W = -\frac{(\Delta x)^2}{2k_A}(\dot{q}_A + \alpha\dot{q}_B) \ .$$

If the conductive interface is parallel to the xz plane, the equation centered at node u_C becomes

$$\frac{K}{\beta}v_N + \frac{(\beta+1)(\Delta y)^2}{\alpha(\alpha+1)(\Delta x)^2}u_E + \frac{(\beta+1)(\Delta y)^2}{\gamma(\gamma+1)(\Delta z)^2}u_U$$

$$-\left(\frac{K+\beta}{\beta} + \frac{(\beta+1)(\Delta y)^2}{\alpha(\Delta x)^2} + \frac{(\beta+1)(\Delta y)^2}{\gamma(\Delta z)^2}\right)u_C \qquad (5.24)$$

$$+\frac{(\beta+1)(\Delta y)^2}{(\gamma+1)(\Delta z)^2}u_L + \frac{(\beta+1)(\Delta y)^2}{(\alpha+1)(\Delta x)^2}u_W + u_S = -\frac{(\Delta y)^2}{2k_A}(\dot{q}_A + \beta\dot{q}_B) \ .$$

If the conductive interface is parallel to the xy plane, the equation centered at node u_C is

$$\frac{K}{\gamma}v_U + \frac{(\gamma+1)(\Delta z)^2}{\beta(\beta+1)(\Delta y)^2}u_N + \frac{(\gamma+1)(\Delta z)^2}{\alpha(\alpha+1)(\Delta x)^2}u_E$$

$$-\left(\frac{K+\gamma}{\gamma} + \frac{(\gamma+1)(\Delta z)^2}{\beta(\Delta y)^2} + \frac{(\gamma+1)(\Delta z)^2}{\alpha(\Delta x)^2}\right)u_C \qquad (5.25)$$

$$+\frac{(\gamma+1)(\Delta z)^2}{(\alpha+1)(\Delta x)^2}u_W + \frac{(\gamma+1)(\Delta z)^2}{(\beta+1)(\Delta y)^2}u_S + u_L = -\frac{(\Delta z)^2}{2k_A}(\dot{q}_A + \gamma\dot{q}_B) \ .$$

Equations 5.23 through 5.25 are used to replace the ELLPACK generated equations for nodes that lie along the conductive interface of a three dimensional regime when non-uniform grid analysis is used. These equations properly model the heat transport between the chip and the substrate in the problem discussed in Chapter I.

2. Example: Reducing the Conductive Interface Equation for a Non-uniform Grid in a Three Dimensional Region to the Seven Point Star Equation

a. Problem Statement

This problem is to show that Equation 5.23 reduces to the seven point star equation if two materials with the same thermal properties come into contact along a conductive interface. We assume that the heat source is uniform throughout the two materials. Uniform grid analysis is used in this example.

b. Data

Using a uniform grid for the analysis implies

$$\begin{aligned} \Delta x = \Delta y = \Delta z = h, \\ \alpha = \beta = \gamma = 1. \end{aligned} \tag{5.26}$$

Since the material properties are the same, the following is true:

$$k_A = k_B = k, \tag{5.27}$$

which implies that

$$K = \frac{k_A}{k_B} = 1. \tag{5.28}$$

Since the heat source is the same throughout both materials it is true that

$$\dot{q}_A = \dot{q}_b = \dot{q}. \tag{5.29}$$

c. Solution

Substituting Equation 5.26 into Equation 5.23 generates

$$Kv_E + u_N + u_U - (K+5)u_C + u_L + u_S + u_W = -\frac{h}{2k_A}(\dot{q}_A + \dot{q}_B) \ . \tag{5.30}$$

Equation 5.30 approximates the behavior at a conductive interface parallel to the yz plane on a uniform grid. Substituting Equations 5.27 through 5.29 into Equation 5.30 yields

$$v_E + u_N + u_U - 6u_C + u_L + u_S + u_W = -\frac{h}{k_A}\dot{q} \ . \tag{5.31}$$

Equation 5.31 is the seven point star equation that was first introduced in Chapter II.

The seven point star equation can also be obtained if Equations 5.26 through 5.29 are substituted into Equation 5.24 or into Equation 5.25. This example lends credibility to the equations derived for heat transfer along the conductive interface in a three dimensional region. These equations are used in an ELLPACK program described in Chapter VI to model the behavior along the interface between the chip and the substrate in the problem of Chapter I.

VI. MODEL FOR A CHIP MOUNTED ON A SUBSTRATE

This chapter describes a program written to approximate the temperature profile within the region depicted in Figure 4.1. The equations developed in Chapters IV and V are used in concert with ELLPACK to model the heat transport within the region. The program allows the user to vary the dimensional and material property parameters of the region with little modification to the code. The formulation of the program utilizes the symmetry of the region as discussed in Chapter I. The resulting solution estimates the temperature for the entire region depicted in Figure 1.3.

A. SUBROUTINES/VARIABLES USED THROUGHOUT THE COMPUTER MODEL

User defined subroutines are incorporated in the SUBPROGRAM segment of the main ELLPACK program. These subroutines and their use are summarized in Appendix A. User defined variables are introduced into the main program and made available to the remainder of the ELLPACK program by means of the GLOBAL segment or through subroutine calls. These variables are listed in Appendix B. Several ELLPACK generated variables are manipulated and changed throughout the program run when equations along convective/conductive interfaces are modified. These variables are listed in Appendix C. The program is listed in its entirety in Appendix D.

B. PROGRAM ALGORITHM

A basic outline for the program algorithm is presented below. The steps listed that are flagged with '***' indicate that the user must make changes if the Appendix B variables are modified.

1. Set the OPTIONS Segment***

Output choices are set in the OPTIONS segment. Invoke the MEMORY or TIME modules if memory allocation or run time data is desired. Set LEVEL equal to 0, 1, or 2 depending on the level of output desired. In addition to output provided by ELLPACK, the following data is printed for each LEVEL setting:

a. LEVEL 0 Output

A summary table for data that is generated by invoking an ELLPACK provided solution module is printed. Additionally, error banners are displayed if the program run is terminated due to improper specification of grid points and interface locations.

b. LEVEL 1 Output

In addition to LEVEL 0 output, LEVEL 1 output generates a banner when any conductive or convective interface routine is invoked to alter the equations along such an interface. LEVEL 1 also summons the user defined successive over-relaxation (SOR) subroutine and provides a summary table of its output. LEVEL 1 output for the test problem is listed in Appendix E.

c. LEVEL 2 Output

LEVEL 2 output is used for debugging. In addition to LEVEL 0 output and LEVEL 1 output, LEVEL 2 output lists the parameters used to modify each equation along the conductive or convective interfaces. LEVEL 2 output lists the values of each coefficient for the equations that are to be modified. LEVEL 2 output also calls attention to any modifications that may have been made due to boundary conditions invoked on the interface. Finally, LEVEL 2 output provides additional output to debug the user provided SOR subroutine. A sample of LEVEL 2 output is provided in Appendix F.

2. Declare User Defined Variables

Variable declarations are made in the DECLARATIONS segment of the main ELLPACK program. A double precision array, UNKN, is declared in the test program for use as a work space if the SOR subroutine is invoked.

3. Establish Global Variables

The GLOBAL segment is used to establish global variables. The dimensions, material properties, heat generation levels, and convective properties for the region depicted in Figure 4.1 are made global for use throughout the program.

4. Define the Elliptic Equation

The elliptic equation is defined in the EQUATION segment. This segment uses double precision functions A, B, and C as variable coefficients to represent the three equations for each subregion depicted in Figure 4.1 as a single ELLPACK equation. These functions automatically make the required changes to the EQUATION segment

when the user varies the dimensions of each subregion. However, they do not adjust the EQUATION segment if major changes to the region's basic geometry are made.

5. Define the Boundary of the Region***

If the user wishes to alter the boundary conditions, changes must be made in the BOUNDARY segment. If the locations of the boundaries are shifted, the user must make appropriate changes to accomplish this in the BOUNDARY and the GRID segments.

6. Initialize Test Problem Variables***

The first FORTRAN segment is used to initialize the values for the dimensions, material properties, and heat source magnitudes of the region illustrated in Figure 4.1. These global variables may be changed to modify the problem and will be automatically adjusted throughout the user defined subroutines and functions.

7. Define the Grid***

The grid points are defined in the GRID segment. The user must ensure that the region's dimensions and interface locations, specified in the first FORTRAN segment, are also specified in the grid. The outlying grid points must also agree with the location of the boundaries specified in the BOUNDARY segment. If this criteria is not met the program will not discretize the equations properly and error routines incorporated in the user defined subroutines will terminate the program execution.

8. Discretize the Elliptic Equation

The elliptic equation is discretized by ELLPACK in the DISCRETIZATION segment. The 7 POINT STAR module will be used exclusively for the three dimensional test problem.

9. Invoke Interface Subroutines

User defined subroutines that modify the difference equations along convective or conductive interfaces are called from the second FORTRAN block. Before calling interface subroutines, the subroutine Q3STRT is invoked to initialize local variables that are used throughout the program. Q3STRT also sets the logical variable FIRST. This variable serves as a flag indicating that the current subroutine call is the first call to a convective subroutine. The FIRST flag is immediately reset after this first call. The program then invokes a set of convective subroutines that define interfaces that intersect along a convex edge or corner. For the test problem the equations along the upper boundary of the substrate are modified first where

$$z = RT2, \qquad 0 \le x \le RL/2, \qquad 0 \le y \le RL/2. \tag{6.1}$$

Note that this subroutine call has incorrectly modified the equations along the conductive interface between the chip and substrate. This will be corrected in a later call to the conductive subroutines.

The remaining convective surfaces form a concave intersection with the upper boundary of the substrate. This necessitates resetting the FIRST flag which is accomplished by invoking Q3STRT again. Immediately thereafter, convective subroutines

that modify equations along the eastern and western boundaries of the chip are respectively invoked at

$$\begin{aligned} x &= RT1H, \quad 0 \le y \le RT2, \quad RT2 \le z \le RT1 + RT2, \\ y &= RT2H, \quad 0 \le x \le RT1, \quad RT2 \le z \le RT1 + RT2. \end{aligned} \tag{6.2}$$

Finally a conductive subroutine that modifies the equations along the interface between the chip and substrate is called at

$$z = RT2, \quad 0 \le x \le RL1, \quad 0 \le y \le RL2. \tag{6.3}$$

This FORTRAN block need not be altered provided that the basic geometry of Figure 4.1 is not changed.

A call to a conductive or convective subroutine summons a series of additional operations that are automatically invoked within each respective subregion.

a. Define the Bounds of the Interface

The physical bounds of each interface are entered by the user in the first FORTRAN segment when defining the dimensions for the subregions Figure 4.1. The values are passed to the conductive or convective subroutines via common blocks. If the user alters the geometry of the problem or changes the names or usage of variables initialized in the first FORTRAN segment, it becomes necessary to alter variables which are used in subroutines Q3EVCT, Q3NVCT, Q3UVCT, Q3WVCT, Q3SVCT, Q3LVCT, and Q3UDCT. If the user merely changes the Appendix B variables, no modifications are required.

b. Define Material Properties on Both Sides of the Interface

The same actions are taken as in Subsection a above.

c. Incorporate Boundary Conditions Along the Interface

If the interface intersects with a boundary that was defined in the BOUNDARY segment, the boundary condition must be incorporated into the modified interface equation. This is accomplished within user defined subroutines. This step requires no modification if the user properly specifies boundaries in the BOUNDARY segment.

10. Invoke the User Defined SOR Subroutine*

This SOR subroutine, Q8SORM, may be invoked by selecting LEVEL 1 output. This subroutine is suggested for large matrix systems. Parameters in this algorithm may be initialized in the first FORTRAN segment and are defined in Appendix B.

11. Select a Solution Module*

The SOLUTION segment may be used to invoke any solution module discussed in Chapter III to solve the test problem. However, if the size of the matrix precludes their use, the user provided SOR algorithm is recommended.

12. Select an Output Module*

If an ELLPACK provided output module is desired, it may be called in the OUTPUT segment. However, the user provided function, Q8SOLN, will provide a summary of data that is more compact and easier to read than ELLPACK data tables.

This summary is automatically invoked for LEVEL 0 output when an ELLPACK solution module is specified. It is also invoked via the SOR subroutine if LEVEL 1 output is requested. ELLPACK plotting modules are restricted to data for two dimensional regions.

C. RUNNING A TEST PROBLEM

A test problem is described using the values summarized in Table 6.1. The problem is solved using both the user defined Q8SORM subroutine and the ELLPACK provided BAND GE module. The LEVEL 1 output for this run is listed in Appendix E. The solution using the SOR algorithm matches the solution obtained using BAND GE to the second decimal place. This is a verification of the SOR algorithm written for this problem.

There is an advantage to using the Q8SORM subroutine over the BAND GE module. The BAND GE module requires

$$(2ML + MU + 2) * N \tag{6.1}$$

words of work space where:

- ML equals the lower bandwidth of the matrix being solved.
- MU equals the upper bandwidth of the matrix.
- N equals the number of equations for the system.

The bandwidth of a matrix resulting from the 7 POINT STAR module is determined by taking the product of the number of x points and y points defined in the GRID segment.

Table 6.1: Values Used For Test Problem.

Subregion	Variable	Value	Units
Chip	RL1	0.002	M
	RL2	0.002	M
	RT1	0.002	M
	RK1	1.0	W/M-K
	QDT1	6.25E7	W/M^3
Substrate	RL	0.05	M
	RT2	0.0005	M
	RK2	50.0	W/M-K
	QDT2	0.0	W/M^3
Fluid Region	H	50.0	W/M^2-K
	UINF	20.0	°C
Convective Interfaces	RL1H	RL1/2	M
	RL2H	RL2/2	M
	RL3H	RT2	M
Conductive Interface	RT2Z	RT2	M

For the test problem the bandwidth is 81. The size of the system is 648 equations. Thus BAND GE requires 158,760 words of storage. The storage requirement for BAND GE rapidly grows beyond memory limits if the grid is refined or if a region of greater complexity or size is defined.

The memory requirement is reduced significantly by iteratively solving the system of equations using the SOR method. For example, the work space requirement for the

Table 6.2: Relaxation Factor versus Number of Iterations Required to Solve the Test Problem when the Convergence Criteria Equal to .0001.

Relaxation Factor	Number of Iterations	Relaxation Factor	Number of Iterations
1.00	7625	1.50	2977
1.05	7007	1.55	2640
1.10	6434	1.60	2317
1.15	5902	1.65	2006
1.20	5406	1.70	1705
1.25	4940	1.75	1413
1.30	4503	1.80	1127
1.35	4091	1.85	843
1.40	3700	1.90	552
1.45	3330	1.95	252

test problem is reduced to 648 words of storage. This is a savings in storage of 98.6 percent! The tradeoff, however, is that SOR is a slower and less accurate algorithm. The speed of solution can be increased, however, by determining the optimum relaxation factor. Table 6.2 summarizes the number of iterations required for relaxation factors between one and two. The optimum relaxation factor was determined to be 1.9445. When used in the test program, the convergence criteria of .0001 is reached in 227 iterations.

The speed of SOR can also be increased by appropriately selecting the value used for convergence criteria. Recall that the error of the numerical approximation of the

temperature profile when using the seven point star algorithm on a non-uniform grid is of the order of the maximum grid interval in the x, y, or z direction. For the test problem this maximum grid interval is .007 meters. It would not be reasonable to set a convergence criteria that is much smaller than this value. A good rule of thumb is to set the convergence criteria equal to the maximum grid length divided by about 100. In the test problem, the convergence criteria is .0001.

Despite efforts to speed up the SOR algorithm, it still runs slower than BAND GE. However, in today's world of high speed computers the delay is not likely to be noticeable to the user. Furthermore, the savings in work space becomes paramount as the size of the system increases. Table 6.3 provides some examples of storage requirements that one might encounter when using the 7 POINT STAR module to approximate the temperature profile in a three dimensional domain. The matrix of size 10000 by 10000

Table 6.3: Storage Requirements for BAND GE versus SOR for Systems of Various Sizes.

Number of Grid Points			Matrix Characteristics		Storage Requirements	
x	y	z	Bandwidth	Size	BAND GE	SOR
3	3	3	9	27	783	27
5	5	5	25	125	9,625	125
9	9	9	81	648	158,760	648
25	25	16	625	10,000	$18*10^6$	10^4

listed for the final system in Table 6.3 is not at all unreasonable for a configuration of greater geometric complexity or increased grid refinement.

From this discussion, it is evident that SOR is the preferable method for solving large systems of equations due to storage limitations in today's computers.

D. A MODEL FOR AN INFINITE NUMBER OF UNIFORM CHIPS MOUNTED ON AN INFINITE SUBSTRATE

Up to this point the temperature profile for a single chip mounted on a substrate has been approximated. A slight modification to the boundary conditions will extend this system to an infinite number of equally spaced uniform chips mounted on an infinite substrate. In this system, the eastern and northern boundaries of the substrate in Figure 4.1 will also be planes of symmetry. Thus at the eastern boundary of the substrate the boundary condition in ELLPACK form is

$$\text{UX} = 0 \qquad \text{X}=.025\text{D}0. \tag{6.5}$$

Likewise, the boundary condition at the northern boundary of the substrate becomes

$$\text{UY} = 0 \qquad \text{Y} = .025\text{D}0. \tag{6.6}$$

Incorporating these boundary conditions into the test problem generates the LEVEL 0 output listed in Appendix G. The resulting approximation of the temperature profile throughout the infinite region is equal within two decimal points to that temperature profile obtained from the problem modeling a single chip on a substrate. The fact that the solutions for the single chip case and the infinite chip case are nearly equal is not

surprising since the cross sectional area of the chip in the test problem is much smaller than the cross sectional area of the substrate. Hence the heat generated by the chip which is transferred via conduction to the substrate has enough space in the substrate to convectively dissipate to the fluid medium before the eastern or northern boundary of the substrate is reached. This result would tend to suggest that the solution obtained from the test problem would also approximate the temperature profile in a finite multi-chip region as depicted in Figure 1.1. That solution would fall somewhere between the single chip case and the infinite chip case.

VII. SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH

The primary purpose of this thesis is to develop the equations that can be used to predict the temperature profile in an electronic complex containing several chips mounted on a substrate with convection to a fluid medium as the primary mode of cooling. The main results, reported in Chapters IV and V, modeled behavior along convective and conductive interfaces respectively. These results are required to modify equations generated by ELLPACK, a FORTRAN based software package whose capabilities are discussed in Chapters II and III. A program is presented to model the temperature profile for a single chip mounted on a substrate. A minor modification to the boundary conditions makes it possible to use this same program to predict the temperature profile in a region where an infinite number of identical chips are evenly distributed on an infinite substrate. The results, indistinguishable to two decimal places, are reported in Appendices E and G respectively. In both the single and infinite cases, these results appear to provide a reasonable approximation of the temperature profile within the electronic complex. For the infinite case the chips were approximately one centimeter apart. It is recommended that further investigation be performed to determine how closely the chips can be packed before the infinite chip case deviates significantly from the single chip case. This investigation could be accomplished by simply modifying the dimensions of the substrate. The solution for the case of a finite number of identical

chips uniformly mounted on a finite substrate is bounded below by the single chip case and above by the infinite chip case.

It would be desirable to further develop the program so that the temperature profile could be predicted for a complex that contains a finite number of disparate heat generating electronic components that are non-uniformly distributed on a finite substrate, e.g. a circuit board. This complex geometry would generate a matrix system of large proportions. The SOR subroutine that has been provided can handle these large systems. The program presented in this paper provides some of the basic building blocks needed to construct such a region and approximate its temperatures. However, several areas still remain to be investigated before this can be accomplished.

The program itself should first be thoroughly tested by varying boundary values and by experimenting with the variables that define the properties and dimensions for the region illustrated in Figure 4.3. The program should then be streamlined in such a way that construction of a complex region similar to the one discussed in the previous paragraph could be reasonably constructed.

Several subroutines still remain to be written to further generalize the program. An output routine that produces three dimensional temperature plots for each z-level defined by the GRID segment would be extremely useful. A conductive routine that alters equations for conductive interfaces lying parallel to the xz and yz plane is still needed. Equations that model the behavior when more than two solid materials come into contact should be developed and coded.

With the advent of parallel processing, one can solve the problem discussed in this paper using domain decomposition which, as the name suggests, breaks a domain down into smaller subdomains. Each subdomain is then assigned to a processor which can use ELLPACK to simultaneously solve the elliptic problem defined over each subdomain. Currently ‖ELLPACK (parallel ELLPACK) is under development at Purdue University. The algorithms prescribed in this paper could be incorporated into ‖ELLPACK to be utilized on those subdomains where a conductive interface is present.

The ultimate goal is to provide a system that will enable the thermal designer to approximate the temperatures within a wide range of electronic configurations. An optimum program could be described as computerized electronic building block set. Each "block" represents an electronic component that is assigned attributes that quantify dimensions, material properties, etc. The "blocks" are placed on a "base board" that represents the substrate. The attributes of the substrate and surrounding fluid medium are also be defined by the user. If this procedure is accomplished using a computer graphics package, the task of building a complex electronic package would be simplified immensely. The user would then execute a program that produces a solution. In this idealized package the steps required to set up an ELLPACK program are performed automatically thus minimizing user intervention.

A software package such as the one described would be invaluable to the thermal designer. This package would be an advanced distant cousin to the program provided in this paper. However, the equations derived here provide the necessary foundation to pursue this final goal.

APPENDIX A

USER PROVIDED SUBROUTINES

Several subroutines were written to be used together with ELLPACK to model the chip and substrate system. These subroutines are identified using ELLPACK naming conventions discussed in Rice and Boisvert [Ref. 4:pp. 317-318]. These subroutines are listed in the order in which they appear in the program.

- Q3STRT: Initializes variables used in the interface discretization routines. These variables are unchanged throughout the program. Also sets FIRST flag that signals a first call to a convective subroutine.
- Q3EVCT: Invokes the subroutines that modify the coefficients for equations generated by ELLPACK along an eastern convective interface.
- Q3NVCT: Same as Q3EVCT along a northern convective interface.
- Q3UVCT: Same as Q3EVCT along an upper convective interface.
- Q3UDCT: Invokes the subroutines that modify the coefficients for equations generated by ELLPACK along a conductive interface parallel to the xy plane.
- Q3XVCT: Modifies the equations generated by ELLPACK for points on a convective interface parallel to the yz plane.
- Q3YVCT: Same as Q3XVCT along a convective interface that is parallel to the xz plane.
- Q3ZVCT: Same as Q3XVCT along a convective interface that is parallel to the xy plane.
- Q3ZDCT: Same as Q3ZVCT along a conductive interface that is parallel to the xy plane.

- Q3PARM: Returns the parameters necessary to modify the coefficients of the equations on the interface of a three dimensional region where a non-uniform grid is used.

- Q3BNDY: Modifies the values of the coefficients for points that are adjacent to a boundary.

- Q3STCF: Modifies the ELLPACK variables that describe the equation along the interface of the three dimensional regime. The values returned are used to solve the system of equations.

- Q4XRNG: Determines the range of the unknown grid points along an interface parallel to the yz plane. Ensures that bounds specified for the interface have also been specified on the grid. If not, the run is terminated.

- Q4YRNG: Same as Q4XRNG on interfaces parallel to the xz plane.

- Q4ZRNG: Same as Q4XRNG on interfaces parallel to the xy plane.

- Q5SORM: Solves the system of equations generated by the discretization routines using successive over-relaxation iterative method.

- Q8CVHD: Prints a banner when a convective interface is invoked.

- Q8CDHD: Prints a banner when a conductive interface is invoked.

- Q8EQHD: Prints a header identifying the equation currently being modified.

- Q8EQDT: Prints parameters used when modifying the current equation and prints resulting equation coefficients.

- Q8BDAT: Prints parameters used when modifying interface equations adjacent to boundaries.

- Q8BRST: Prints parameters used when a boundary convective interface.

- Q8SOLN: Produces a set of solution summary tables for each z grid point defined for the region.

APPENDIX B

USER DEFINED VARIABLES

Several user defined variables are introduced and made global for use throughout the entire program. These variable are set by the user in the first FORTRAN block. They include:

- H: The convection coefficient between the chip, substrate and fluid region.
- ITMAX: The maximum iterations attempted for the SOR algorithm.
- OMEGA: The relaxation factor for the SOR algorithm.
- QDT1: The magnitude of the heat source of the chip.
- QDT2: The magnitude of the heat source of the substrate.
- RK1: The thermal conductivity coefficient of the chip.
- RK2: The thermal conductivity of the substrate.
- RL: The length and width of the substrate.
- RL1: The length of the chip.
- RL1H: The location of the convective interface along the eastern boundary of the chip.
- RL2: The width of the chip.
- RL2H: The location of the convective interface along the northern boundary of the chip.
- RL3H: The location of the convective interface along the upper boundary of the substrate.

- RT1: The thickness of the chip.
- RT2: The thickness of the substrate.
- RT2Z: The location of the conductive interface between the chip and the substrate.
- UINF: The ambient temperature of the fluid region.
- ZETA: The convergence criteria for the SOR algorithm.

APPENDIX C

ELLPACK VARIABLES AND SUBROUTINES USED IN THE PROGRAM

This appendix describes the ELLPACK generated variables and subroutines that are used throughout the program to modify equations along the interfaces. A more detailed discussion can be found in Rice and Boisvert [Ref 4:pp. 319-342].

- I1BCTY: A vector of length six that contains integer values that classify the boundary type. Possible values for I1BCTY are summarized below.

Value of I1BCTY	Meaning
1	Dirichlet boundary condition.
2	Neumann boundary condition.
3	Mixed boundary condition.
4	Periodic boundary condition.

- I1IDCO: Column indices for matrix coefficients in R1COEF.
- I1LEVL: Level of printed output requested in the OPTIONS segment.
- I1MNEQ: Row dimension of R1COEF and I1IDCO representing the number of equations and unknown variables generated by the DISCRETIZATION segment.
- I1MNCO: Column dimension of R1COEF and I1DCO representing the maximum number of unknown variables per equation.
- I1NGRX: Number of x grid points listed in the GRID segment.
- I1NGRY: Number of y grid points listed in the GRID segment.

- I1NGRZ: Number of z grid points listed in the GRID segment.
- Q1ERRH: Subroutine that produces a banner indicated that a fatal error has been encountered.
- Q1ERRT: Subroutine that terminates the program if a fatal error has been encountered.
- Q1BCOE: Subroutine used to identify coefficients of a boundary. This routines are automatically set up by ELLPACK based on data placed in the BOUNDARY segment.
- R1AXGR: Initial x grid point listed in the GRID segment.
- R1AYGR: Initial y grid point listed in the GRID segment.
- R1AZGR: Initial z grid point listed in the GRID segment.
- R1BBBB: Matrix containing right hand side values of the equations generated by the DISCRETIZATION segment.
- R1BRHS: Function that generates the values for the right hand side of the boundary conditions which were specified in the BOUNDARY segment.
- R1BXGR: Final x grid point listed in the GRID segment.
- R1BYGR: Final y grid point listed in the GRID segment.
- R1BZGR: Final z grid point listed in the GRID segment.
- R1COEF: An array of non-zero matrix elements that represent the coefficients for each equation generated by the DISCRETIZATION segment.
- R1GRDX: Vector containing x grid points defined in the GRID segment.
- R1GRDY: Vector containing y grid points defined in the GRID segment.
- R1GRDZ: Vector containing z grid points defined in the GRID segment.
- R1UNKN: Vector containing the solution of matrix system.

APPENDIX D

ELLPACK PROGRAM CODE

```
*************************************************************************
*                                                                       *
*   SAMPLE OF 3-D ELLIPTIC PROBLEM USING ELLPACK                        *
*                                                                       *
*        THE FOLLOWING ELLPACK PROGRAM SOLVES A STEADY STATE            *
*        CONDUCTION PROBLEM IN 3-D FOR A RECTANGULAR DOMAIN.            *
*                                                                       *
*TOP BOX OF DIMENSION      -RL <X<RL , -RL <Y<RL , RT <Z<RT +RT        *
*                             1     1     2     2    2     1   2        *
*BOTTOM BOX OF DIMENSION   -RL <X<RL , -RL <Y<RL , 0  <Z<RT            *
*                                                      2                *
*                                                                       *
*************************************************************************

*************************************************************************
*                                                                       *
*   OPTIONS SEGMENT -- SET LEVEL OF DESIRED OUTPUT HERE.  IN ADDITION   *
*   TO STANDARD OUTPUT GENERATED BY ELLPACK, THE FOLLOWING DATA IS      *
*   IS ALSO GENERATED:                                                  *
*                                                                       *
*      LEVEL=0 -- SOLUTION SUMMARY TABLE IS SOLUTION MODULE USED.       *
*      LEVEL=1 -- BANNERS INDICATING THAT VARIOUS DISCRETIZATION        *
*                 AND SOLUTION SUBROUTINES HAVE BEEN INVOKED.           *
*                 ADDITIONALLY PROVIDES A TABLE SUMMARIZING THE         *
*                 SOLUTION TO THE PROBLEM.                              *
*      LEVEL=3 -- A LISTING OF ALL EQUATION COEFFICIENTS AND INDICES    *
*                 THAT HAVE BEEN ALTERED BY CONVECTIVE, CONDUCTIVE,     *
*                 BOUNDARY, AND BOUNDARY RESET ROUTINES.  ALSO LISTS    *
*                 PARAMETERS THAT ARE USED TO MODIFY THE EQUATIONS.     *
*                                                                       *
*************************************************************************

OPTION.        LEVEL=1    $     MEMORY  $  TIME

DECLARATIONS.
               DOUBLE PRECISION UNKN(1:10000)
```

```
*************************************************************************
*                                                                       *
*   GLOBAL SEGMENT -- ALL BOX DIMENSIONS, INTERFACE LOCATIONS, AND      *
*   MATERIAL PROPERTIES ARE MADE GLOBAL FOR SUBSEQUENT USE IN           *
*   USER DEFINED SUBROUTINES.                                           *
*                                                                       *
*************************************************************************

GLOBAL.
              COMMON / BOXDIM / RL,   RL1, RL2,  RT1,  RT2
              COMMON / CINTFC / RL1H, RL2H, RL3H, RT2Z
              COMMON / MATPRP / RK1,  QDT1, RK2, QDT2, H, UINF

*************************************************************************
*                                                                       *
*   EQUATION SEGMENT -- EQUATION MUST BE IN SELF-ADJOINT FORM TO SAT-   *
*   ISFY LIMITATIONS OF THE 7 PT STAR DISCRETIZATION SEGMENT.           *
*                                                                       *
*************************************************************************

EQUATION.     (A(X,Y,Z)*UX)X + (A(X,Y,Z)*UY)Y + (A(X,Y,Z)*UZ)Z +&
               B(X,Y,Z)*U    =  C(X,Y,Z)

BOUNDARY.

* EASTERN  BOUNDARY.
*            RK2*UX +  H*U = H*UINF       ON X = .025D0
                UX         = 0.D0         ON X = .025D0
* NORTHERN BOUNDARY.
*            RK2*UY +  H*U = H*UINF       ON Y = .025D0
                UY         = 0.D0         ON Y = .025D0
* UPPER    BOUNDARY.
             RK1*UZ +  H*U = H*UINF       ON Z = .0025D0
* WESTERN  BOUNDARY.
                UX         = 0.D0         ON X = 0.0D0
* SOUTHERN BOUNDARY.
                UY         = 0.D0         ON Y = 0.0D0
* LOWER    BOUNDARY.
             RK2*UZ - H*U  = -H*UINF      ON Z = 0.0D0
```

```
************************************************************************
*                                                                      *
*   FORTRAN SEGMENT 1 -- USE TO ENTER THE DIMENSIONS OF THE CHIP       *
*   AND SUBSTRATE AND THE LOCATION OF ALL INTERFACES.                  *
*   ALSO ENTERED HERE ARE THE THERMAL PROPERTIES AND MAGNITUDES        *
*   OF HEAT SOURCES FOR ALL MATERIALS IN THE REGION.                   *
*                                                                      *
************************************************************************

FORTRAN.
C
C ENTER DIMENSIONS OF CHIP AND SUBSTRATE.
C
              RL   = .05D0
              RL1  = .002D0
              RL2  = .002D0
              RT1  = .002D0
              RT2  = .0005D0
C
C ENTER LOCATION OF CONVECTIVE INTERFACE.
C
              RL1H = .001D0
              RL2H = .001D0
              RL3H = RT2
C
C ENTER LOCATION OF CONDUCTIVE INTERFACE.
C
              RT2Z = RT2
C
C ENTER PROPERTIES OF ELECTRONIC CHIP.
C
              RK1  = 50.D0
              QDT1 = .5D0/(RL1*RL2*RT1)
C
C ENTER PROPERTIES OF SUBSTRATE.
C
              RK2  = 1.D0
              QDT2 = 0.D0/(RL*RL*RT2)
C
C ENTER COEFFICIENTS ALONG CONVECTIVE INTERFACES.
C
              H    = 50.D0
              UINF = 20.D0
C
C  INITIALIZE DATA FOR SOR ROUTINE.
C
              ZETA  = .0001D0
              ITMAX = 500
              OMEGA = 1.9445D0
```

```
**************************************************************************
*                                                                        *
*   GRID SEGMENT --  ENSURE THAT ALL BOX DIMENSIONS AND INTERFACE        *
*   LOCATIONS SPECIFIED IN FORTRAN SEGMENT ONE ARE INCLUDED IN THE       *
*   GRID POINT SPECIFICATION.  THIS PROGRAM WILL TERMINATE IF THIS       *
*   CONDITION IS NOT MET.                                                *
*                                                                        *
**************************************************************************

GRID.          9  X POINTS 0.D0,.0003D0,.0006D0,.001D0,.002D0,.006D0,&
                           .012D0,.018D0,.025D0
               9  Y POINTS 0.D0,.0003D0,.0006D0,.001D0,.002D0,.006D0,&
                           .012D0,.018D0,.025D0
               8  Z POINTS 0.D0,.0001D0,.0003D0,.0005D0,.0007D0,&
                           .0011D0,.0018D0,.0025D0

DISCRET.       7 POINT 3D

**************************************************************************
*                                                                        *
*   FORTRAN SEGMENT 2 -- THIS SEGMENT IS USED TO INVOKE THE SUB-         *
*   ROUTINES THAT ARE REQUIRED TO MODIFY ELLPACK GENERATED EQUATIONS     *
*   ALONG THE MATERIAL INTERFACES.  ADDITIONALLY IT WILL INVOKE ANY      *
*   SOLUTION OR OUTPUT ROUTINES THAT HAVE BEEN SPECIFICALLY WRITTEN      *
*   FOR THIS PROGRAM.                                                    *
*                                                                        *
**************************************************************************

FORTRAN.

C
C SET INITIAL GLOBAL VALUES BEFORE INVOKING CONVECTIVE INTERFACES.
C
      CALL Q3STRT
C
C INVOKE UPPER CONVECTIVE INTERFACE AT Z = RT2.
C
      CALL Q3UVCT(R1COEF, R1BBBB, I1IDCO, I1MNEQ, I1MNCO)
C
C RESET INITIAL GLOBAL VALUES BEFORE INVOKING CONVECTIVE INTERFACES
C SINCE REMAINING CONVECTIVE INTERFACES ARE CONCAVE TO FIRST INTERFACE.
C
      CALL Q3STRT
C
C INVOKE EASTERN CONVECTIVE INTERFACE AT X = RL1H.
C
      CALL Q3EVCT(R1COEF, R1BBBB, I1IDCO, I1MNEQ, I1MNCO)
C
C INVOKE NORTHERN CONVECTIVE INTERFACE AT Y = RL2H.
C
      CALL Q3NVCT(R1COEF, R1BBBB, I1IDCO, I1MNEQ, I1MNCO)
C
C INVOKE Z CONDUCTIVE INTERFACE AT Z = RT2Z.

C
      CALL Q3UDCT(R1COEF, R1BBBB, I1IDCO, I1MNEQ, I1MNCO)
C
C IF LEVEL 1 OUTPUT REQUESTED, EXECUTE SOR ALGORITHM.
C
      IF (I1LEVL .GE. 1)
```

```
A    CALL Q5SORM(R1BBBB,R1COEF,I1IDCO,OMEGA,ZETA,ITMAX,UNKN,I1MNEQ,
B                I1MNCO)

SOLUTION.      BAND GE

FORTRAN.
               CALL Q8SOLN(R1UNKN)
```

SUBPROGRAMS.

```
      DOUBLE PRECISION FUNCTION A(X,Y,Z)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C FUNCTION A ASSIGNS VALUES FOR THE UXX, UYY, AND UZZ COEFFICIENTS IN THE
C ELLIPTIC EQUATION.  VALUES ARE BASED ON THE LOCATION OF THE INTERFACES.
C
      COMMON / BOXDIM / RL,   RL1, RL2, RT1,  RT2
      COMMON / CINTFC / RL1H, RL2H, RL3H, RT2Z
      A = 1.D0
      IF(Z .LE. RT2)  GOTO 10
      IF(X .GT. RL1H) A = 0.D0
      IF(Y .GT. RL2H) A = 0.D0
10    RETURN
      END


      DOUBLE PRECISION FUNCTION B(X,Y,Z)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C FUNCTION B ASSIGNS VALUES FOR THE U COEFFICIENT IN THE ELLIPTIC
C EQUATION.  VALUES ARE PREDICATED ON THE LOCATION OF THE INTERFACES.
C
      COMMON / BOXDIM / RL,   RL1, RL2, RT1,  RT2
      COMMON / CINTFC / RL1H, RL2H, RL3H, RT2Z
      B = 0.D0
      IF(Z .LE. RT2)  GOTO 10
      IF(X .GT. RL1H) B = 1.D0
      IF(Y .GT. RL2H) B = 1.D0
10    RETURN
      END


      DOUBLE PRECISION FUNCTION C(X,Y,Z)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C FUNCTION C ASSIGNS VALUES FOR THE RIGHT HAND SIDE IN THE ELLIPTIC
C EQUATION.  VALUES ARE PREDICATED ON THE LOCATION OF THE INTERFACES.
C
      COMMON / BOXDIM / RL,   RL1,   RL2, RT1,  RT2
      COMMON / CINTFC / RL1H, RL2H, RL3H, RT2Z
      COMMON / MATPRP / RK1,  QDT1, RK2, QDT2,   H, UINF
      C = QDT2/RK2
      IF(Z .LE. RT2)  GOTO 10
      IF(X .GT. RL1H) C = UINF
      IF(Y .GT. RL2H) C = UINF
      IF(X .LE. RL1H .AND. Y .LE. RL2H) C = -QDT1/RK1
10    RETURN
      END
```

```
      SUBROUTINE Q3STRT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E  PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q3STRT INITIALIZES VARIABLES USED IN THE INTERFACE DISCRETIZATION
C   THAT REMAIN UNCHANGED THROUGHOUT THE PROGRAM RUN.  THESE INCLUDE;
C
C        I1,J1,K1 -- GRID POSITIONS OF THE FIRST UNKNOWN X, Y, & Z NODE
C        I2,J2,K2 -- GRID POSITIONS OF THE  LAST UNKNOWN X, Y, & Z NODE
C        NX,NY,NZ -- NUMBER OF UNKNOWN X, Y, & Z NODES ON THE GRID
C        FIRST    -- LOGICAL VARIABLE SET TO TRUE FOR FIRST CONVECTIVE
C                    RUN.  RESET TO FALSE THEREAFTER.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   MARCH 1991
C
C-----------------------------------------------------------------------
C
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
      COMMON / C1BCTY / I1BCTY(1)
      COMMON / ISTART/  I1, I2, NX, J1, J2, NY, K1, K2, NZ, FIRST
      LOGICAL FIRST
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS,ISDIRC
     A       /    1,    2,    3,    4,    5,    6,     7,  8,     1 /
C
C NUMBER OF X GRID POINTS ON THE 3-D DOMAIN FOR WHICH U IS UNKNOWN.
C
      I1 = 1
      IF (I1BCTY(WEST) .EQ. ISDIRC) I1 = 2
      I2 = I1NGRX
      IF (I1BCTY(EAST) .EQ. ISDIRC) I2 = I2-1
      NX = I2 - I1 + 1
C
C NUMBER OF Y GRID POINTS ON THE 3-D DOMAIN FOR WHICH U IS UNKNOWN.
C
      J1 = 1
      IF (I1BCTY(SOUTH) .EQ. ISDIRC) J1 = 2
      J2 = I1NGRY
      IF (I1BCTY(NORTH) .EQ. ISDIRC) J2 = J2-1
      NY = J2 - J1 + 1
C
C NUMBER OF Z GRID POINTS ON THE 3-D DOMAIN FOR WHICH U IS UNKNOWN.
C
      K1 = 1
      IF (I1BCTY(LOWER) .EQ. ISDIRC) K1 = 2
      K2 = I1NGRZ
      IF (I1BCTY(UPPER) .EQ. ISDIRC) K2 = K2-1
      NZ = K2 - K1 + 1
C
C SET THE START FLAG.
```

```
C
      FIRST = .TRUE.
      RETURN
      END
```

```
      SUBROUTINE Q3EVCT(COEF, BBBB, IDCO, MNEQ, MNCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C Q3EVCT INVOKES THE SUBROUTINES THAT WILL MODIFY THE COEFFICIENTS
C GENERATED BY  M O D U L E  7 POINT 3D ALONG AN EASTERN CONVECTIVE
C INTERFACE (I.E. FLUID REGION TO THE EAST OF SOLID REGION).
C
C   INPUTS  FOR Q3EVCT ARE:
C
C        X     -- LOCATION OF INTERFACE (I.E. POSITION ON X AXIS).
C        Y1,Y2 -- Y BOUNDS OF INTERFACE (I.E. Y1 <= Y <= Y2).
C        Z1,Z2 -- Z BOUNDS OF INTERFACE (I.E. Z1 <= Z <= Z2).
C        Q     -- INTERNAL HEAT GENERATION VALUE OF THE SOLID MATERIAL.
C        RK    -- THERMAL CONDUCTIVITY OF THE SOLID MATERIAL.
C        TINF  -- AMBIENT TEMPERATURE OF THE FLUID MATERIAL.
C        H     -- CONVECTION COEFFICIENT ALONG THE SOLID/FLUID INTERFACE.
C
      COMMON / C1IVCN / I1LEVL,I1INPT,I1OUTP,I1SCRA,I1KWRK,I1KORD
      COMMON / BOXDIM / RL,   RL1,   RL2, RT1,  RT2
      COMMON / CINTFC / RL1H, RL2H, RL3H, RT2Z
      COMMON / MATPRP / RK1,  QDT1,  RK2,  QDT2,   H, UINF
      DOUBLE PRECISION COEF(MNEQ,MNCO), BBBB(MNEQ)
      INTEGER          IDCO(MNEQ,MNCO)
C
C ENTER MATERIAL PROPERTY DATA.
C
      Q    = QDT1
      RK   = RK1
      TINF = UINF
C
C ENTER LOCATION AND BOUNDS OF CONVECTIVE INTERFACE.
C
      X  = RL1H
      Y1 = 0.D0
      Y2 = RL2H
      Z1 = RT2
      Z2 = RT1+RT2
C
C PRINT HEADER REPORTING THAT CONVECTIVE SUBROUTINE WAS INVOKED.
C
      IF (I1LEVL .GT. 0)
     ACALL Q8CVHD(Q,RK,TINF,H,'X',X,Y1,Y2,Z1,Z2,'EASTERN ')
C
C MODIFY THE COEFFICIENTS ALONG THE INTERFACE.
C
      CALL Q3XVCT(X, Y1, Y2, Z1, Z2, Q, RK, TINF, H, 1,
     A            COEF, BBBB, IDCO, MNEQ, MNCO)
      RETURN
      END


      SUBROUTINE Q3NVCT(COEF, BBBB, IDCO, MNEQ, MNCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C Q3NVCT INVOKES THE SUBROUTINES THAT WILL MODIFY THE COEFFICIENTS
C GENERATED BY  M O D U L E  7 POINT 3D ALONG A NORTHERN CONVECTIVE
C INTERFACE (I.E. FLUID REGION TO THE NORTH OF SOLID REGION).
C
C   INPUTS  FOR Q3NVCT ARE:
C
```

```
C        X1,X2 -- X BOUNDS OF INTERFACE (I.E. X1 <= X <= X2).
C        Y     -- LOCATION OF INTERFACE (I.E. POSITION ON Y AXIS).
C        Z1,Z2 -- Z BOUNDS OF INTERFACE (I.E. Z1 <= Z <= Z2).
C        Q     -- INTERNAL HEAT GENERATION VALUE OF THE SOLID MATERIAL.
C        RK    -- THERMAL CONDUCTIVITY OF THE SOLID MATERIAL.
C        TINF  -- AMBIENT TEMPERATURE OF THE FLUID MATERIAL.
C        H     -- CONVECTION COEFFICIENT ALONG THE SOLID/FLUID INTERFACE.
C
      COMMON / C1IVCN / I1LEVL,I1INPT,I1OUTP,I1SCRA,I1KWRK,I1KORD
      COMMON / BOXDIM / RL,   RL1,   RL2, RT1,  RT2
      COMMON / CINTFC / RL1H, RL2H, RL3H, RT2Z
      COMMON / MATPRP / RK1,  QDT1,  RK2,  QDT2,   H, UINF
      DOUBLE PRECISION COEF(MNEQ,MNCO), BBBB(MNEQ)
      INTEGER          IDCO(MNEQ,MNCO)
C
C ENTER MATERIAL PROPERTY DATA.
C
      Q    = QDT1
      RK   = RK1
      TINF = UINF
C
C ENTER LOCATION AND BOUNDS OF CONVECTIVE INTERFACE.
C
      Y  =  RL2H
      X1 = 0.D0
      X2 = RL1H
      Z1 = RT2
      Z2 = RT1+RT2
C
C PRINT HEADER REPORTING THAT CONVECTIVE SUBROUTINE WAS INVOKED.
C
      IF (I1LEVL .GT. 0)
     ACALL Q8CVHD(Q,RK,TINF,H,'Y',Y,X1,X2,Z1,Z2,'NORTHERN')
C
C MODIFY THE COEFFICIENTS ALONG THE INTERFACE.
C
      CALL Q3YVCT(Y, X1, X2, Z1, Z2, Q, RK, TINF, H, 1,
     A            COEF, BBBB, IDCO, MNEQ, MNCO)
      RETURN
      END
```

```
      SUBROUTINE Q3UVCT(COEF, BBBB, IDCO, MNEQ, MNCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C Q3UVCT INVOKES THE SUBROUTINES THAT WILL MODIFY THE COEFFICIENTS
C GENERATED BY  M O D U L E  7 POINT 3D ALONG AN UPPER CONVECTIVE
C INTERFACE (I.E. FLUID REGION ABOVE THE SOLID REGION).
C
C   INPUTS  FOR Q3UVCT ARE:
C
C        X1,X2 -- X BOUNDS OF INTERFACE (I.E. X1 <= X <= X2).
C        Y1,Y2 -- Y BOUNDS OF INTERFACE (I.E. Y1 <= Y <= Y2).
C        Z     -- LOCATION OF INTERFACE (I.E. POSITION ON Z AXIS).
C        QDT1  -- INTERNAL HEAT GENERATION VALUE OF THE SOLID MATERIAL.
C        RK    -- THERMAL CONDUCTIVITY OF THE SOLID MATERIAL.
C        TINF  -- AMBIENT TEMPERATURE OF THE FLUID MATERIAL.
C        H     -- CONVECTION COEFFICIENT ALONG THE SOLID/FLUID INTERFACE.
C
      COMMON / C1IVCN / I1LEVL,I1INPT,I1OUTP,I1SCRA,I1KWRK,I1KORD
      COMMON / BOXDIM / RL,   RL1,   RL2, RT1,  RT2
      COMMON / CINTFC / RL1H, RL2H, RL3H, RT2Z
      COMMON / MATPRP / RK1,  QDT1,  RK2,  QDT2,   H, UINF
      DOUBLE PRECISION COEF(MNEQ,MNCO), BBBB(MNEQ)
      INTEGER          IDCO(MNEQ,MNCO)
C
C ENTER MATERIAL PROPERTY DATA.
C
      Q    = QDT2
      RK   = RK2
      TINF = UINF
C
C ENTER LOCATION AND BOUNDS OF CONVECTIVE INTERFACE.
C
      Z  = RL3H
      X1 = 0.D0
      X2 = RL/2.D0
      Y1 = 0.D0
      Y2 = RL/2.D0
C
C PRINT HEADER REPORTING THAT CONVECTIVE SUBROUTINE WAS INVOKED.
C
      IF (I1LEVL .GT. 0)
     ACALL Q8CVHD(Q,RK,TINF,H,'Z',Z,X1,X2,Y1,Y2,'UPPER   ')
C
C MODIFY THE COEFFICIENTS ALONG THE INTERFACE
C
      CALL Q3ZVCT(Z, X1, X2, Y1, Y2, Q, RK, TINF, H, 1,
     A            COEF, BBBB, IDCO, MNEQ, MNCO)
      RETURN
      END
```

```
      SUBROUTINE Q3UDCT(COEF, BBBB, IDCO, MNEQ, MNCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C Q3UDCT INVOKES THE SUBROUTINES THAT WILL MODIFY THE COEFFICIENTS
C GENERATED BY  M O D U L E  7 POINT 3D ALONG A CONDUCTIVE INTERFACE
C THAT IS PARALLEL TO THE XY PLANE.
C
C DATA VARIABLES FOR Q3UDCT ARE:
C
C     RKA/RKB -- THERMAL CONDUCTIVITY COEFFICIENT ON MATERIAL BELOW/
C                ABOVE THE Z INTERFACE.
C     QA/QB   -- HEAT GENERATION COEFFICIENT OF MATERIAL BELOW/ABOVE THE
C                Z INTERFACE.
C     X1,X2   -- X BOUNDS FOR Z CONDUCTIVE INTERFACE.
C     Y1,Y2   -- Y BOUNDS FOR Z CONDUCTIVE INTERFACE.
C     Z       -- POINT AT WHICH Z INTERFACE INTERSECTS Z AXIS.
C
      COMMON / C1IVCN / I1LEVL,I1INPT,I1OUTP,I1SCRA,I1KWRK,I1KORD
      COMMON / BOXDIM / RL,   RL1,   RL2, RT1,  RT2
      COMMON / CINTFC / RL1H, RL2H, RL3H, RT2Z
      COMMON / MATPRP / RK1,  QDT1,  RK2,  QDT2,   H, UINF
      DOUBLE PRECISION COEF(MNEQ,MNCO), BBBB(MNEQ)
      INTEGER          IDCO(MNEQ,MNCO)
C
C ENTER MATERIAL PROPERTY DATA.
C
      QA  = QDT2
      RKA = RK2
      QB  = QDT1
      RKB = RK1
C
C ENTER LOCATION AND BOUNDS OF CONDUCTIVE INTERFACE.
C
      Z = RT2
      X1 = 0.D0
      X2 = RL1H
      Y1 = 0.D0
      Y2 = RL2H
C
C PRINT HEADER REPORTING THAT CONDUCTIVE SUBROUTINE WAS INVOKED.
C
      IF (I1LEVL .GT. 0)
     ACALL Q8CDHD(QA,RKA,QB,RKB,'Z',Z,X1,X2,Y1,Y2)
C
C MODIFY THE COEFFICIENTS ALONG THE INTERFACE.
C
      CALL Q3ZDCT(Z, X1, X2, Y1, Y2, QA, RKA, QB,RKB,
     A            COEF, BBBB, IDCO, MNEQ, MNCO)
      RETURN
      END
```

```
      SUBROUTINE Q3XVCT(X, Y1, Y2, Z1, Z2, Q, RK, TINF, H, IDIR,
     A                  COEF, BBBB, IDCO, MNEQ, MNCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E  PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q3XVCT MODIFIES THE DISCRETE EQUATIONS GENERATED BY  M O D U L E
C   7 POINT 3D FOR POINTS LYING ON A CONVECTIVE PHASE INTERFACE THAT
C   IS PARALLEL TO THE XZ PLANE IN A 3 DIMENSIONAL RECTANGULAR REGION.
C
C   INPUTS AND LOCAL VARIABLES FOR Q3XVCT ARE:
C
C        X     -- LOCATION OF INTERFACE (I.E. POSITION ON X AXIS).
C        Y1,Y2 -- Y BOUNDS OF INTERFACE (I.E. Y1 <= Y <= Y2).
C        Z1,Z2 -- Z BOUNDS OF INTERFACE (I.E. Z1 <= Z <= Z2).
C        Q     -- INTERNAL HEAT GENERATION VALUE OF THE SOLID MATERIAL.
C        RK    -- THERMAL CONDUCTIVITY OF THE SOLID MATERIAL.
C        TINF  -- AMBIENT TEMPERATURE OF THE FLUID MATERIAL.
C        H     -- CONVECTION COEFFICIENT ALONG THE SOLID/FLUID INTERFACE.
C        IDIR  -- DIRECTION OF FLUID REGION IN RELATION TO THE INTERFACE.
C                 (IDIR = 1 MEANS FLUID REGION TO THE NORTH OF THE
C                 SOLID REGION.  IDIR = -1 MEANS FLUID REGION TO THE
C                 SOUTH OF THE SOLID REGION.)
C        L3CVCT-- ARRAY OF LOGICAL ELEMENTS SPECIFYING IF A POINT LIES
C                 ON ONE OR MORE CONVECTIVE INTERFACES.
C        L3BNDY-- ARRAY OF LOGICAL ELEMENTS SPECIFYING IF A POINT LIES ON
C        NEQN  -- INDEX OF EQUATION CURRENTLY BEING MODIFIED.
C        CO    -- TEMP. ARRAY CONTAINING EQUATION COEFFICIENT VALUES.
C        OLDCO -- TEMP ARRAY CONTAINING PREVIOUS EQN. COEFFICIENT VALUES.
C                 A BOUNDARY OR IS OR ADJACENT TO A DIRICHLET BOUNDARY.
C
C   E L L P A C K  VARIABLES USED AS INPUTS FOR Q3XVCT ARE:
C
C        MNEQ  -- NUMBER OF EQUATIONS IN THE SYSTEM.
C        MNCO  -- NUMBER OF COEFFICIENTS PER EQUATION.
C        COEF  -- COEFFICIENT FOR EACH VARIABLE OF EACH EQUATION.
C        BBBB  -- RIGHT HAND SIDE VALUE OF EACH EQUATION.
C        IDCO  -- INTEGER SUBSCRIPT THAT IDENTIFIES POSITION OF THE
C                 VARIABLE WITHIN THE GRID.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   FEBRUARY 1991
C
C  N O T E
C
C   USES JUNE 1978/SEPTEMBER 1982 VERSION OF 7 POINT 3D  M O D U L E
C   WRITTEN BY ROGER GRIMES.
C
C-----------------------------------------------------------------------
C
      COMMON / C1IVCN / I1LEVL,I1INPT,I1OUTP,I1SCRA,I1KWRK,I1KORD
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
```

```
      COMMON / C1GRDX / R1GRDX(1)
      COMMON / C1GRDY / R1GRDY(1)
      COMMON / C1GRDZ / R1GRDZ(1)
      COMMON / ISTART/  I1, I2, NX, J1, J2, NY, K1, K2, NZ, FIRST
      COMMON CO, OLDCO
      DOUBLE  PRECISION CO(1:10000,8),OLDCO(1:10000,8)
      COMMON  L3CVCT, L3BNDY
      LOGICAL L3CVCT(1:10000,6),L3BNDY(1:10000,6), FIRST
      DOUBLE PRECISION  COEF(MNEQ, MNCO)
      DOUBLE PRECISION  BBBB(MNEQ)
      INTEGER IDCO(MNEQ, MNCO)
      CHARACTER*8 POSIT(8)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS,ISDIRC
     A       /    1,    2,    3,    4,    5,    6,    7,  8,     1 /
C
C IF VARIABLE DIMENSIONS OF L3CVCT, L3BNDY, CO, OLDCO ARE EXCEEDED
C TERMINATE PROGRAM.
C
      IF (MNEQ .GT. 10000) THEN
           CALL Q1ERRH(17HSUBROUTINE Q3XVCT,17)
           WRITE(6,3)
    3      FORMAT(/'MAXIMUM DIMENSIONS OF ARRAYS L3CVCT, L3BNDY, CO,'/
     A     'AND OLDCO HAVE BEEN EXCEEDED.  MODIFY DIMENSION '/
     B     'SPECIFICATION OR REDUCE THE TOTAL NUMBER OF GRID POINTS.'/)
           CALL Q1ERRT
           RETURN
      ENDIF
C
C IF THIS IS THE FIRST CALL OF A CONVECTION INTERFACE SUBROUTINE THEN
C RESET CONVECTIVE INTERFACE FLAGS AND RESET FIRST FLAG.
C
      IF (FIRST) THEN
           DO 5 I = 1, MNEQ
                DO 5 J = 1, 6
                     L3CVCT(I,J) = .FALSE.
                     L3BNDY(I,J) = .FALSE.
    5      CONTINUE
      ENDIF
      FIRST = .FALSE.
C
C DETERMINE THE GRID LOCATION OF THE Y INTERFACE AND THE RANGE OF
C THE X AND Z BOUNDS.
C
      CALL Q4XRNG(X,Y1,Y2,Z1,Z2,IX,JY1,JY2,KZ1,KZ2)
      I = IX
      DO 30 K = KZ1, KZ2
           DO 30 J = JY1, JY2
C
C DETERMINE THE NUMBER OF THE EQUATION THAT IS TO BE MODIFIED.
C
              NEQN = (I-I1+1) + NX*((J-J1) + NY*(K-K1))
C
C TEMPORARY DATA HEADER TO DISPLAY RELEVANT PARAMETERS FOR DEBUGGING.
C
              IF (I1LEVL .GT. 1) CALL Q8EQHD(NEQN)
C
C CALCULATE THE PARAMETERS AT THE CURRENT GRID POINT THAT WILL BE USED
C FOR MODIFYING THE EQUATION COEFFICIENTS ON THE INTERFACE.
C
```

```
              CALL Q3PARM(I,J,K,X,DX,A,Y,DY,B,Z,DZ,C)
C
C MODIFY THE COEFFICIENTS OF THE EQUATIONS AT THE INTERFACE.
C
              OLDDX2 = DX**2.D0
              IF (IDIR .EQ. -1) DX = R1GRDX(IX+1) - R1GRDX(IX)
              DX2 = DX**2.D0
              DY2 = DY**2.D0
              DZ2 = DZ**2.D0
C
C *************** EVALUATE CENTER AND RHS COEFFICIENTS ***************
C IF THE CENTER AND RIGHT HAND SIDE COEFFICIENTS HAVE NOT BEEN PREVIOUSLY
C INITIALIZED BY A CONVECTIVE SUBROUTINE, THEN INITIALIZE THEM.  IF THEY
C THEY HAVE BEEN INITIALIZED THEN MODIFY THEIR VALUES.
C
              IF (L3CVCT(NEQN,  EAST)  .OR. L3CVCT(NEQN, SOUTH).OR.
     A            L3CVCT(NEQN,  WEST)  .OR. L3CVCT(NEQN, NORTH) .OR.
     B            L3CVCT(NEQN, UPPER)  .OR. L3CVCT(NEQN, LOWER)) THEN
                     CO(NEQN,CENTER) = CO(NEQN,CENTER) - H/(DX*RK)
     A                                  + 1.D0/(A*OLDDX2) - 1.D0/DX2
                     CO(NEQN,RHS)    =  CO(NEQN,RHS) - H*TINF/(RK*DX)
              ELSE
                     CO(NEQN,CENTER) = - (H*DX/RK+1.D0)/DX2
     A                                  - 1.D0/(B*DY2) - 1.D0/(C*DZ2)
                     CO(NEQN,RHS)    = - Q/(2.D0*RK)  - H*TINF/(RK*DX)
              ENDIF
C
C ************* EVALUATE NORTHERN AND SOUTHERN COEFFICIENTS *************
C IF SOUTH OR NORTH COEFFICIENTS HAVE NOT BEEN SET BY A PREVIOUS CON-
C VECTIVE SUBROUTINE THEN INITIALIZE THEIR VALUES, OTHERWISE DO NOTHING.
C
              IF (L3CVCT(NEQN,SOUTH) .OR.
     A            L3CVCT(NEQN,NORTH)) GOTO 15
                     CO(NEQN,SOUTH) = 1.D0/((B+1.D0)*DY2)
                     CO(NEQN,NORTH) = CO(NEQN,SOUTH)/B
C
C ************* EVALUATE EASTERN AND WESTERN COEFFICIENTS ************
C IT THE Q3XVCT IS INVOKING A WESTERN CONVECTIVE INTERFACE, THEN
C SET THE WESTERN CONVECTIVE INTERFACE FLAG AND INITIALIZE VALUES FOR
C WESTERN COEFFICIENT.
C
   15         IF (IDIR .EQ. -1) THEN
                   CO(NEQN,EAST) = 1.D0/DX2
                   CO(NEQN,WEST) = 0.D0
                   L3CVCT(NEQN,WEST) = .TRUE.
              ENDIF
C
C IT THE Q3XVCT IS INVOKING A EASTERN CONVECTIVE INTERFACE, THEN
C SET THE EASTERN CONVECTIVE INTERFACE FLAG AND INITIALIZE VALUES FOR
C EASTERN COEFFICIENT.
C
              IF (IDIR .EQ.  1) THEN
                   CO(NEQN,WEST) = 1.D0/DX2
                   CO(NEQN,EAST) = 0.D0
                   L3CVCT(NEQN,EAST) = .TRUE.
              ENDIF
C
C ************* EVALUATE  UPPER AND LOWER  COEFFICIENTS **********
C IF UPPER OR LOWER COEFFICIENTS HAVE NOT BEEN SET BY A PREVIOUS
C CONVECTIVE SUBROUTINE INITIALIZE THEIR VALUES, OTHERWISE DO NOTHING.
```

```
C
                IF (L3CVCT(NEQN,LOWER) .OR.
     A              L3CVCT(NEQN,UPPER)) GOTO 20
                CO(NEQN,LOWER) = 1.D0/((C+1.D0)*DZ2)
                CO(NEQN,UPPER) = CO(NEQN,LOWER)/C
C
C ********************  INVOKE BOUNDARY CONDITIONS  **********************
C MODIFY COEFFICIENTS IF THE GRID POINT IS ADJACENT TO A BOUNDARY.
C
   20           IF ((I .LE. 2) .OR. (I .GE. (I1NGRX-1)) .OR.
     A              (J .LE. 2) .OR. (J .GE. (I1NGRY-1)) .OR.
     B              (K .LE. 2) .OR. (K .GE. (I1NGRZ-1)))THEN
                CALL Q3BNDY(I,J,K,NEQN,X,Y,Z,DX,DY,DZ,L3BNDY,CO,OLDCO)
                ENDIF
C
C ******************  MODIFY  E L L P A C K  VARIABLES  ******************
C
                CALL Q3STCF(COEF,BBBB,IDCO,NEQN,L3CVCT,L3BNDY,MNEQ,MNCO,
     A                      CO)
C
C *************  TEMPORARY OUTPUT STATEMENTS FOR DEBUGGING ***************
C
       IF (I1LEVL .GT. 1) THEN
          A = 1.D0
          IF (L3CVCT(NEQN,SOUTH)) THEN
             B  = 1.D0
             DY = R1GRDY(J+1) - R1GRDY(J)
          ENDIF
          IF (L3CVCT(NEQN,NORTH)) B  = 1.D0
          IF (L3CVCT(NEQN,LOWER)) THEN
             C  = 1.D0
             DZ = R1GRDZ(K+1) - R1GRDZ(K)
          ENDIF
          IF (L3CVCT(NEQN,UPPER)) C  = 1.D0
          CALL Q8EQDT(L3CVCT,NEQN,I,X,DX,A,NX,I1,  0,  0,J,Y,DY,B,NY,J1,
     A          JY1,JY2,K,Z,DZ,C,NZ,K1,KZ1,KZ2,COEF,IDCO,BBBB,MNEQ,MNCO)
       ENDIF
   30  CONTINUE
C
C SAVE CO VALUES GENERATED BY THIS CONVECTIVE INTERFACE SUBROUTINE FOR
C USE IN ANY CALLS TO CONVECTIVE INTERFACE SUBROUTINES THAT MAY FOLLOW.
C
       I = IX
       DO 40 K = KZ1, KZ2
            DO 40 J = JY1, JY2
                 DO 40 N = 1,8
                      NEQN = (I-I1+1) + NX*((J-J1) + NY*(K-K1))
                      OLDCO(NEQN,N) = CO(NEQN,N)
   40  CONTINUE
       RETURN
       END
```

```
      SUBROUTINE Q3YVCT(Y, X1, X2, Z1, Z2, Q, RK, TINF, H, IDIR,
     A                  COEF, BBBB, IDCO, MNEQ, MNCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E  PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q3YVCT MODIFIES THE DISCRETE EQUATIONS GENERATED BY  M O D U L E
C   7 POINT 3D FOR A POINT LYING ON A CONVECTIVE PHASE INTERFACE THAT
C   IS PARALLEL TO THE XZ PLANE IN A 3 DIMENSIONAL RECTANGULAR REGION.
C
C   INPUTS AND LOCAL VARIABLES FOR Q3YVCT ARE:
C
C        X1,X2 -- X BOUNDS OF INTERFACE (I.E. X1 <= X <= X2).
C        Y     -- LOCATION OF INTERFACE (I.E. POSITION ON Y AXIS).
C        Z1,Z2 -- Z BOUNDS OF INTERFACE (I.E. Z1 <= Z <= Z2).
C        Q     -- INTERNAL HEAT GENERATION VALUE OF THE SOLID MATERIAL.
C        RK    -- THERMAL CONDUCTIVITY OF THE SOLID MATERIAL.
C        TINF  -- AMBIENT TEMPERATURE OF THE FLUID MATERIAL.
C        H     -- CONVECTION COEFFICIENT ALONG THE SOLID/FLUID INTERFACE.
C        IDIR  -- POSITION OF FLUID REGION WITH RESPECT TO THE INTERFACE.
C                 (IDIR = 1 MEANS FLUID REGION TO THE NORTH OF THE
C                 SOLID REGION.  IDIR = -1 MEANS FLUID REGION TO THE
C                 SOUTH OF THE SOLID REGION.)
C        L3CVCT-- ARRAY OF LOGICAL ELEMENTS SPECIFYING IF A POINT LIES
C                 ON ONE OR MORE CONVECTIVE INTERFACES.
C        L3BNDY-- ARRAY OF LOGICAL ELEMENTS SPECIFYING IF A POINT LIES ON
C                 A BOUNDARY OR IS OR ADJACENT TO A DIRICHLET BOUNDARY.
C        NEQN  -- INDEX OF EQUATION CURRENTLY BEING MODIFIED.
C        CO    -- TEMP. ARRAY CONTAINING EQUATION COEFFICIENT VALUES.
C        OLDCO -- TEMP ARRAY CONTAINING PREVIOUS EQN. COEFFICIENT VALUES.
C
C   E L L P A C K  VARIABLES USED AS INPUTS FOR Q3YVCT ARE:
C
C        MNEQ   - NUMBER OF EQUATIONS IN THE SYSTEM.
C        MNCO   - NUMBER OF COEFFICIENTS PER EQUATION.
C        COEF   - COEFFICIENT FOR EACH VARIABLE OF EACH EQUATION.
C        BBBB   - RIGHT HAND SIDE VALUE OF EACH EQUATION.
C        IDCO   - INTEGER SUBSCRIPT THAT IDENTIFIES POSITION OF THE
C                 VARIABLE WITHIN THE GRID.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   FEBRUARY 1991
C
C  N O T E
C
C   USES JUNE 1978/SEPTEMBER 1982 VERSION OF 7 POINT 3D  M O D U L E
C   WRITTEN BY ROGER GRIMES.
C
C-----------------------------------------------------------------------
C
      COMMON / C1IVCN / I1LEVL,I1INPT,I1OUTP,I1SCRA,I1KWRK,I1KORD
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
```

```
      COMMON / C1GRDX / R1GRDX(1)
      COMMON / C1GRDY / R1GRDY(1)
      COMMON / C1GRDZ / R1GRDZ(1)
      COMMON / ISTART/  I1, I2, NX, J1, J2, NY, K1, K2, NZ, FIRST
      COMMON  CO, OLDCO
      DOUBLE  PRECISION CO(1:10000,8), OLDCO(1:10000,8)
      COMMON  L3CVCT, L3BNDY
      LOGICAL L3CVCT(1:10000,6),L3BNDY(1:10000,6),FIRST
      DOUBLE PRECISION  COEF(MNEQ, MNCO)
      DOUBLE PRECISION  BBBB(MNEQ)
      INTEGER IDCO(MNEQ, MNCO)
      CHARACTER*8 POSIT(8)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS,ISDIRC
     A       /    1,    2,    3,    4,    5,    6,    7,  8,     1 /
C
C IF VARIABLE DIMENSIONS OF L3CVCT, L3BNDY, CO, OLDCO ARE EXCEEDED
C TERMINATE PROGRAM.
C
      IF (MNEQ .GT. 10000) THEN
           CALL Q1ERRH(17HSUBROUTINE Q3YVCT,17)
           WRITE(6,3)
    3      FORMAT(/'MAXIMUM DIMENSIONS OF ARRAYS L3CVCT, L3BNDY, CO,'/
     A     'AND OLDCO HAVE BEEN EXCEEDED.  MODIFY DIMENSION '/
     B     'SPECIFICATION OR REDUCE THE TOTAL NUMBER OF GRID POINTS.'/)
           CALL Q1ERRT
           RETURN
      ENDIF
C
C IF THIS IS THE FIRST CALL OF A CONVECTION INTERFACE SUBROUTINE THEN
C RESET CONVECTIVE INTERFACE FLAGS AND RESET FIRST FLAG.
C
      IF (FIRST) THEN
           DO 5 I = 1, MNEQ
                DO 5 J = 1, 6
                     L3CVCT(I,J) = .FALSE.
                     L3BNDY(I,J) = .FALSE.
    5      CONTINUE
      ENDIF
      FIRST = .FALSE.
C
C DETERMINE THE GRID LOCATION OF THE Y INTERFACE AND THE RANGE OF
C THE X AND Z BOUNDS.
C
      CALL Q4YRNG(Y,X1,X2,Z1,Z2,JY,IX1,IX2,KZ1,KZ2)
      J = JY
      DO 30 K = KZ1, KZ2
           DO 30 I = IX1, IX2
C
C DETERMINE THE NUMBER OF THE EQUATION THAT IS TO BE MODIFIED
C
              NEQN = (I-I1+1) + NX*((J-J1) + NY*(K-K1))
C
C TEMPORARY DATA HEADER TO DISPLAY RELEVANT PARAMETERS FOR DEBUGGING
C
              IF (I1LEVL .GT. 1) CALL Q8EQHD(NEQN)
C
C CALCULATE THE PARAMETERS AT THE CURRENT GRID POINT THAT WILL BE USED
C FOR MODIFYING THE EQUATION COEFFICIENTS ON THE INTERFACE
C
```

```
              CALL Q3PARM(I,J,K,X,DX,A,Y,DY,B,Z,DZ,C)
C
C MODIFY THE COEFFICIENTS OF THE EQUATIONS AT THE INTERFACE
C
              OLDDY2 = DY**2.D0
              IF (IDIR .EQ. -1) DY = R1GRDY(JY+1) - R1GRDY(JY)
              DX2 = DX**2.D0
              DY2 = DY**2.D0
              DZ2 = DZ**2.D0
C
C *************** EVALUATE CENTER AND RHS COEFFICIENTS ***************
C IF THE CENTER AND RIGHT HAND SIDE COEFFICIENTS HAVE NOT BEEN PREVIOUSLY
C INITIALIZED BY A CONVECTIVE SUBROUTINE, THEN INITIALIZE THEM.  IF THEY
C THEY HAVE BEEN INITIALIZED THEN MODIFY THEIR VALUES.
C
              IF (L3CVCT(NEQN,  EAST)  .OR. L3CVCT(NEQN, SOUTH).OR.
     A            L3CVCT(NEQN,  WEST)  .OR. L3CVCT(NEQN, NORTH) .OR.
     B            L3CVCT(NEQN, UPPER)  .OR. L3CVCT(NEQN, LOWER)) THEN
                     CO(NEQN,CENTER) = CO(NEQN,CENTER) - H/(DY*RK)
     A                                  + 1.D0/(B*OLDDY2) - 1.D0/DY2
                     CO(NEQN,RHS)    = CO(NEQN,RHS)   - H*TINF/(RK*DY)
              ELSE
                     CO(NEQN,CENTER) = - (H*DY/RK+1.D0)/DY2
     A                                  - 1.D0/(A*DX2) - 1.D0/(C*DZ2)
                     CO(NEQN,RHS)    = - Q/(2.D0*RK)  - H*TINF/(RK*DY)
              ENDIF
C
C ************* EVALUATE EASTERN AND WESTERN COEFFICIENTS *************
C IF WEST OR EAST COEFFICIENT HAVE NOT BEEN SET BY A PREVIOUS CONVECTIVE
C SUBROUTINE THEN INITIALIZE THEIR VALUES, OTHERWISE DO NOTHING.
C
              IF (L3CVCT(NEQN,WEST) .OR.
     A            L3CVCT(NEQN,EAST)) GOTO 15
                     CO(NEQN,WEST) = 1.D0/((A+1.D0)*DX2)
                     CO(NEQN,EAST) = CO(NEQN,WEST)/A
C
C ************* EVALUATE NORTHERN AND SOUTHERN COEFFICIENTS *************
C IF Q3YVCT IS INVOKING A SOUTHERN CONVECTIVE INTERFACE, THEN
C SET THE SOUTHERN CONVECTIVE INTERFACE FLAG AND INITIALIZE VALUES FOR
C NORTH AND SOUTH COEFFICIENTS.
C
   15         IF (IDIR .EQ. -1) THEN
                   CO(NEQN,NORTH) = 1.D0/DY2
                   CO(NEQN,SOUTH) = 0.D0
                   L3CVCT(NEQN,SOUTH) = .TRUE.
              ENDIF
C
C IF Q3YVCT IS INVOKING A NORTHERN CONVECTIVE INTERFACE, THEN
C SET THE NORTHERN CONVECTIVE INTERFACE FLAG AND INITIALIZE VALUES FOR
C NORTH AND SOUTH COEFFICIENTS.
C
              IF (IDIR .EQ.  1) THEN
                   CO(NEQN,SOUTH) = 1.D0/DY2
                   CO(NEQN,NORTH) = 0.D0
                   L3CVCT(NEQN,NORTH) = .TRUE.
              ENDIF
C
C ************* EVALUATE  UPPER AND LOWER  COEFFICIENTS *************
C IF UPPER OR LOWER COEFFICIENTS HAVE NOT BEEN SET BY A PREVIOUS
C CONVECTIVE SUBROUTINE INITIALIZE THEIR VALUES, OTHERWISE DO NOTHING.
```

```
C
              IF (L3CVCT(NEQN,LOWER) .OR.
     A            L3CVCT(NEQN,UPPER)) GOTO 20
              CO(NEQN,LOWER) = 1.D0/((C+1.D0)*DZ2)
              CO(NEQN,UPPER) = CO(NEQN,LOWER)/C
C
C ********************  INVOKE BOUNDARY CONDITIONS  ********************
C MODIFY COEFFICIENTS IF THE GRID POINT IS ADJACENT TO A BOUNDARY.
C
   20         IF ((I .LE. 2) .OR. (I .GE. (I1NGRX-1)) .OR.
     A             (J .LE. 2) .OR. (J .GE. (I1NGRY-1)) .OR.
     B             (K .LE. 2) .OR. (K .GE. (I1NGRZ-1)))
     C         CALL Q3BNDY(I,J,K,NEQN,X,Y,Z,DX,DY,DZ,L3BNDY,CO,OLDCO)
C
C ******************  MODIFY  E L L P A C K  VARIABLES  ******************
C
              CALL Q3STCF(COEF,BBBB,IDCO,NEQN,L3CVCT,L3BNDY,MNEQ,MNCO,
     A                    CO)
C
C *************  TEMPORARY OUTPUT STATEMENTS FOR DEBUGGING ***************
C
      IF (I1LEVL .GT. 1) THEN
      B = 1.D0
      IF (L3CVCT(NEQN, WEST)) THEN
          A  = 1.D0
          DX = R1GRDX(I+1) - R1GRDX(I)
      ENDIF
      IF (L3CVCT(NEQN, EAST)) A  = 1.D0
      IF (L3CVCT(NEQN,LOWER)) THEN
          C  = 1.D0
          DZ = R1GRDZ(K+1) - R1GRDZ(K)
      ENDIF
      IF (L3CVCT(NEQN,UPPER)) C  = 1.D0
      CALL Q8EQDT(L3CVCT,NEQN,I,X,DX,A,NX,I1,IX1,IX2,J,Y,DY,B,NY,J1,
     A          0,  0,K,Z,DZ,C,NZ,K1,KZ1,KZ2,COEF,IDCO,BBBB,MNEQ,MNCO)
      ENDIF
   30 CONTINUE
C
C SAVE CO VALUES GENERATED BY THIS CONVECTIVE INTERFACE SUBROUTINE FOR
C USE IN ANY CALLS TO CONVECTIVE INTERFACE SUBROUTINES THAT MAY FOLLOW.
C
      J = JY
      DO 40 K = KZ1, KZ2
           DO 40 I = IX1, IX2
                DO 40 N = 1,8
                     NEQN = (I-I1+1) + NX*((J-J1) + NY*(K-K1))
                     OLDCO(NEQN,N) = CO(NEQN,N)
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE Q3ZVCT(Z, X1, X2, Y1, Y2, Q, RK, TINF, H, IDIR,
     A                  COEF, BBBB, IDCO, MNEQ, MNCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C---------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E  PHASE 3D
C---------------------------------------------------------------------
C  P U R P O S E
C
C   Q3ZVCT MODIFIES THE DISCRETE EQUATIONS GENERATED BY  M O D U L E
C   7 POINT 3D FOR A POINT LYING ON A CONVECTIVE PHASE INTERFACE THAT
C   IS PARALLEL TO THE XY PLANE IN A 3 DIMENSIONAL RECTANGULAR REGION.
C
C   INPUTS AND LOCAL VARIABLES FOR Q3ZVCT ARE:
C
C        X1,X2 -- X BOUNDS OF INTERFACE (I.E. X1 <= X <= X2).
C        Y1,Y2 -- Y BOUNDS OF INTERFACE (I.E. Y1 <= Y <= Y2).
C        Z     -- LOCATION OF INTERFACE (I.E. POSITION ON Z AXIS).
C        Q     -- INTERNAL HEAT GENERATION VALUE OF THE SOLID MATERIAL.
C        RK    -- THERMAL CONDUCTIVITY OF THE SOLID MATERIAL.
C        TINF  -- AMBIENT TEMPERATURE OF THE FLUID MATERIAL.
C        H     -- CONVECTION COEFFICIENT ALONG THE SOLID/FLUID INTERFACE.
C        IDIR  -- POSITION OF FLUID REGION WITH RESPECT TO THE INTERFACE.
C                 (IDIR = 1 MEANS FLUID REGION ABOVE THE SOLID REGION.
C                 IDIR = -1 MEANS FLUID REGION  BELOW THE SOLID REGION.)
C        L3CVCT-- ARRAY OF LOGICAL ELEMENTS SPECIFYING IF A POINT LIES
C                 ON ONE OR MORE CONVECTIVE INTERFACES.
C        L3BNDY-- ARRAY OF LOGICAL ELEMENTS SPECIFYING IF A POINT LIES ON
C                 A BOUNDARY OR IS OR ADJACENT TO A DIRICHLET BOUNDARY.
C        NEQN  -- INDEX OF EQUATION CURRENTLY BEING MODIFIED.
C        CO    -- TEMP. ARRAY CONTAINING EQUATION COEFFICIENT VALUES.
C        OLDCO -- TEMP ARRAY CONTAINING PREVIOUS EQN. COEFFICIENT VALUES.
C
C   E L L P A C K  VARIABLES USED AS INPUTS FOR Q3ZVCT ARE:
C
C        MNEQ  -- NUMBER OF EQUATIONS IN THE SYSTEM.
C        MNCO  -- NUMBER OF COEFFICIENTS PER EQUATION.
C        COEF  -- COEFFICIENT FOR EACH VARIABLE OF EACH EQUATION.
C        BBBB  -- RIGHT HAND SIDE VALUE OF EACH EQUATION.
C        IDCO  -- INTEGER SUBSCRIPT THAT IDENTIFIES POSITION OF THE
C                 VARIABLE WITHIN THE GRID.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   FEBRUARY 1991
C
C  N O T E
C
C   USES JUNE 1978/SEPTEMBER 1982 VERSION OF 7 POINT 3D  M O D U L E
C   WRITTEN BY ROGER GRIMES.
C
C---------------------------------------------------------------------
C
      COMMON / C1IVCN / I1LEVL,I1INPT,I1OUTP,I1SCRA,I1KWRK,I1KORD
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
      COMMON / C1GRDX / R1GRDX(1)
```

```
      COMMON / C1GRDY / R1GRDY(1)
      COMMON / C1GRDZ / R1GRDZ(1)
      COMMON / ISTART/  I1, I2, NX, J1, J2, NY, K1, K2, NZ, FIRST
      COMMON  CO, OLDCO
      DOUBLE PRECISION CO(1:10000,8), OLDCO(1:10000,8)
      COMMON  L3CVCT, L3BNDY
      LOGICAL L3CVCT(1:10000,6),L3BNDY(1:10000,6), FIRST
      DOUBLE  PRECISION  COEF(MNEQ, MNCO)
      DOUBLE  PRECISION  BBBB(MNEQ)
      INTEGER IDCO(MNEQ, MNCO)
      CHARACTER*8 POSIT(8)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS,ISDIRC
     A       /    1,    2,    3,    4,    5,    6,      7,  8,     1 /
C
C IF VARIABLE DIMENSIONS OF L3CVCT, L3BNDY, CO, OLDCO ARE EXCEEDED
C TERMINATE PROGRAM.
C
      IF (MNEQ .GT. 10000) THEN
           CALL Q1ERRH(17HSUBROUTINE Q3ZVCT,17)
           WRITE(6,3)
    3      FORMAT(/'MAXIMUM DIMENSIONS OF ARRAYS L3CVCT, L3BNDY, CO,'/
     A     'AND OLDCO HAVE BEEN EXCEEDED.  MODIFY DIMENSION '/
     B     'SPECIFICATION OR REDUCE THE TOTAL NUMBER OF GRID POINTS.'/)
           CALL Q1ERRT
           RETURN
      ENDIF
C
C IF THIS IS THE FIRST CALL OF A CONVECTION INTERFACE SUBROUTINE THEN
C RESET CONVECTIVE INTERFACE FLAGS AND RESET FIRST FLAG.
C
      IF (FIRST) THEN
           DO 5 I = 1, MNEQ
                DO 5 J = 1, 6
                     L3CVCT(I,J) = .FALSE.
                     L3BNDY(I,J) = .FALSE.
    5      CONTINUE
      ENDIF
      FIRST = .FALSE.
C
C DETERMINE THE GRID LOCATION OF THE Z INTERFACE AND THE RANGE OF
C THE X AND Y BOUNDS.
C
      CALL Q4ZRNG(Z,X1,X2,Y1,Y2,KZ,IX1,IX2,JY1,JY2)
      K = KZ
      DO 30 I = IX1, IX2
           DO 30 J = JY1, JY2
C
C DETERMINE THE NUMBER OF THE EQUATION THAT IS TO BE MODIFIED.
C
              NEQN = (I-I1+1) + NX*((J-J1) + NY*(K-K1))
C
C TEMPORARY DATA HEADER TO DISPLAY RELEVANT PARAMETERS FOR DEBUGGING.
C
              IF (I1LEVL .GT. 1) CALL Q8EQHD(NEQN)
C
C CALCULATE THE PARAMETERS AT THE CURRENT GRID POINT THAT WILL BE USED
C FOR MODIFYING THE EQUATION COEFFICIENTS ON THE INTERFACE.
C
              CALL Q3PARM(I,J,K,X,DX,A,Y,DY,B,Z,DZ,C)
```

```
C
C MODIFY THE COEFFICIENTS OF THE EQUATIONS AT THE INTERFACE.
C
              OLDDZ2 = DZ**2.D0
              IF (IDIR .EQ. -1) DZ = R1GRDZ(KZ+1) - R1GRDZ(KZ)
              DX2 = DX**2.D0
              DY2 = DY**2.D0
              DZ2 = DZ**2.D0
C
C *************** EVALUATE CENTER AND RHS COEFFICIENTS ***************
C IF THE CENTER AND RIGHT HAND SIDE COEFFICIENTS HAVE NOT BEEN PREVIOUSLY
C INITIALIZED BY A CONVECTIVE SUBROUTINE, THEN INITIALIZE THEM.  IF THEY
C THEY HAVE BEEN INITIALIZED THEN MODIFY THEIR VALUES.
C
              IF (L3CVCT(NEQN,  EAST)  .OR. L3CVCT(NEQN, SOUTH).OR.
     A            L3CVCT(NEQN,  WEST)  .OR. L3CVCT(NEQN, NORTH) .OR.
     B            L3CVCT(NEQN, UPPER)  .OR. L3CVCT(NEQN, LOWER)) THEN
                     CO(NEQN,CENTER) = CO(NEQN,CENTER) - H/(DZ*RK)
     A                                  + 1.D0/(C*OLDDZ2) - 1.D0/DZ2
                     CO(NEQN,RHS)    = CO(NEQN,RHS)   - H*TINF/(RK*DZ)
              ELSE
                     CO(NEQN,CENTER) = - (H*DZ/RK+1.D0)/DZ2
     A                                  - 1.D0/(A*DX2) - 1.D0/(B*DY2)
                     CO(NEQN,RHS)    = - Q/(2.D0*RK)  - H*TINF/(RK*DZ)
              ENDIF
C
C ************* EVALUATE EASTERN AND WESTERN COEFFICIENTS *************
C IF WEST OR EAST COEFFICIENT HAVE NOT BEEN SET BY A PREVIOUS CONVECTIVE
C SUBROUTINE THEN INITIALIZE THEIR VALUES, OTHERWISE DO NOTHING.
C
              IF (L3CVCT(NEQN,WEST) .OR.
     A            L3CVCT(NEQN,EAST)) GOTO 15
                     CO(NEQN,WEST) = 1.D0/((A+1.D0)*DX2)
                     CO(NEQN,EAST) = CO(NEQN,WEST)/A
C
C ************* EVALUATE UPPER AND LOWER  COEFFICIENTS *************
C IF Q3ZVCT IS INVOKING A LOWER CONVECTIVE INTERFACE, SET THE LOWER CON-
C VECTIVE INTERFACE FLAG AND INITIALIZE VALUES FOR UPPER AND LOWER
C COEFFICIENTS.
C
   15         IF (IDIR .EQ. -1) THEN
                   CO(NEQN,UPPER) = 1.D0/DZ2
                   CO(NEQN,LOWER) = 0.D0
                   L3CVCT(NEQN,LOWER) = .TRUE.
              ENDIF
C
C IF Q3ZVCT IS INVOKING AN UPPER CONVECTIVE INTERFACE, THEN
C SET THE UPPER CONVECTIVE INTERFACE FLAG AND INITIALIZE VALUES FOR
C UPPER AND LOWER COEFFICIENTS.
C
              IF (IDIR .EQ.  1) THEN
                   CO(NEQN,LOWER) = 1.D0/DZ2
                   CO(NEQN,UPPER) = 0.D0
                   L3CVCT(NEQN,UPPER) = .TRUE.
              ENDIF
C
C ************* EVALUATE NORTHERN AND SOUTHERN COEFFICIENTS *************
C IF NORTH OR SOUTH COEFFICIENTS HAVE NOT BEEN SET BY A PREVIOUS
C CONVECTIVE SUBROUTINE INITIALIZE THEIR VALUES, OTHERWISE DO NOTHING.
C
```

```
            IF (L3CVCT(NEQN,SOUTH) .OR.
     A          L3CVCT(NEQN,NORTH)) GOTO 20
            CO(NEQN,SOUTH) = 1.D0/((B+1.D0)*DY2)
            CO(NEQN,NORTH) = CO(NEQN,SOUTH)/B
C
C ********************  INVOKE BOUNDARY CONDITIONS  **********************
C MODIFY COEFFICIENTS IF THE GRID POINT IS ADJACENT TO A BOUNDARY.
C
   20       IF ((I .LE. 2) .OR. (I .GE. (I1NGRX-1)) .OR.
     A          (J .LE. 2) .OR. (J .GE. (I1NGRY-1)) .OR.
     B          (K .LE. 2) .OR. (K .GE. (I1NGRZ-1)))
     C       CALL Q3BNDY(I,J,K,NEQN,X,Y,Z,DX,DY,DZ,L3BNDY,CO,OLDCO)
C
C ******************  MODIFY  E L L P A C K  VARIABLES  ******************
C
            CALL Q3STCF(COEF,BBBB,IDCO,NEQN,L3CVCT,L3BNDY,MNEQ,MNCO,
     A                  CO)
C
C *************  TEMPORARY OUTPUT STATEMENTS FOR DEBUGGING ***************
C
      IF (I1LEVL .GT. 1) THEN
      C = 1.D0
      IF (L3CVCT(NEQN, WEST)) THEN
          A  = 1.D0
          DX = R1GRDX(I+1) - R1GRDX(I)
      ENDIF
      IF (L3CVCT(NEQN, EAST)) A  = 1.D0
      IF (L3CVCT(NEQN,SOUTH)) THEN
          B  = 1.D0
          DY = R1GRDY(J+1) - R1GRDY(J)
      ENDIF
      IF (L3CVCT(NEQN,NORTH)) B  = 1.D0
      CALL Q8EQDT(L3CVCT,NEQN,I,X,DX,A,NX,I1,IX1,IX2,J,Y,DY,B,NY,J1,
     A       JY1,JY2,K,Z,DZ,C,NZ,K1,  0,  0,COEF,IDCO,BBBB,MNEQ,MNCO)
      ENDIF
   30 CONTINUE
C
C SAVE CO VALUES GENERATED BY THIS CONVECTIVE INTERFACE SUBROUTINE FOR
C USE IN ANY CALLS TO CONVECTIVE INTERFACE SUBROUTINES THAT MAY FOLLOW.
C
      K = KZ
      DO 40 I = IX1, IX2
           DO 40 J = JY1, JY2
                DO 40 N = 1,8
                     NEQN = (I-I1+1) + NX*((J-J1) + NY*(K-K1))
                     OLDCO(NEQN,N) = CO(NEQN,N)
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE Q3ZDCT(Z,X1,X2,Y1,Y2,QA,RKA,QB,RKB,
     A                  COEF,BBBB,IDCO,MNEQ,MNCO)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION   M O D U L E   PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q3ZDCT MODIFIES THE DISCRETE EQUATIONS GENERATED BY  M O D U L E
C   7 POINT 3D FOR A POINT LYING ON A CONDUCTIVE INTERFACE THAT IS
C   PARALLEL TO THE XY PLANE IN A 3 DIMENSIONAL RECTANGULAR REGION.
C
C DATA VARIABLES FOR Q3ZDCT ARE:
C
C     RKA/RKB -- THERMAL CONDUCTIVITY COEFFICIENT ON MATERIAL BELOW/
C                ABOVE THE Z INTERFACE.
C     QA/QB   -- HEAT GENERATION COEFFICIENT OF MATERIAL BELOW/ABOVE THE
C                Z INTERFACE.
C     X1,X2   -- X BOUNDS FOR Z CONDUCTIVE INTERFACE.
C     Y1,Y2   -- Y BOUNDS FOR Z CONDUCTIVE INTERFACE.
C     Z       -- POINT AT WHICH Z INTERFACE INTERSECTS Z AXIS.
C     NEQN    -- INDEX OF EQUATION CURRENTLY BEING MODIFIED.
C     CO      -- TEMP. ARRAY CONTAINING EQUATION COEFFICIENT VALUES.
C     OLDCO   -- TEMP ARRAY CONTAINING PREVIOUS EQN. COEFFICIENT VALUES.
C
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   MARCH 1991
C
C  N O T E
C
C   USES JUNE 1978/SEPTEMBER 1982 VERSION OF 7 POINT 3D  M O D U L E
C   WRITTEN BY ROGER GRIMES.
C
C-----------------------------------------------------------------------
C
      COMMON / C1IVCN / I1LEVL,I1INPT,I1OUTP,I1SCRA,I1KWRK,I1KORD
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
      COMMON / C1GRDX / R1GRDX(1)
      COMMON / C1GRDY / R1GRDY(1)
      COMMON / C1GRDZ / R1GRDZ(1)
      COMMON / ISTART/  I1, I2, NX, J1, J2, NY, K1, K2, NZ, FIRST
      COMMON  CO, OLDCO
      DOUBLE  PRECISION CO(1:10000,8), OLDCO(1:10000,8)
      COMMON  L3CVCT, L3BNDY
      LOGICAL L3CVCT(1:10000,6),L3BNDY(1:10000,6), FIRST
      DOUBLE  PRECISION  COEF(MNEQ, MNCO)
      DOUBLE  PRECISION  BBBB(MNEQ)
      INTEGER IDCO(MNEQ, MNCO)
      CHARACTER*8 POSIT(8)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS,ISDIRC
     A       /    1,    2,   3,    4,    5,    6,     7,  8,     1 /
```

```
C
C IF VARIABLE DIMENSIONS OF L3CVCT, L3BNDY, CO, OLDCO ARE EXCEEDED
C TERMINATE PROGRAM.
C
      IF (MNEQ .GT. 10000) THEN
           CALL Q1ERRH(17HSUBROUTINE Q3ZDCT,17)
           WRITE(6,3)
    3      FORMAT(/'MAXIMUM DIMENSIONS OF ARRAYS L3CVCT, L3BNDY, CO,'/
     A     'AND OLDCO HAVE BEEN EXCEEDED.  MODIFY DIMENSION '/
     B     'SPECIFICATION OR REDUCE THE TOTAL NUMBER OF GRID POINTS.'/)
           CALL Q1ERRT
           RETURN
      ENDIF
C
C RESET CONVECTIVE INTERFACE FLAGS.
C
      DO 5 I = 1, MNEQ
           DO 5 J = 1, 6
                L3CVCT(I,J) = .FALSE.
                L3BNDY(I,J) = .FALSE.
    5 CONTINUE
C
C DETERMINE THE GRID LOCATION OF THE Z INTERFACE AND THE RANGE OF
C THE X AND Y BOUNDS.
C
      CALL Q4ZRNG(Z,X1,X2,Y1,Y2,KZ,IX1,IX2,JY1,JY2)
      K = KZ
      DO 30 I = IX1, IX2
           DO 30 J = JY1, JY2
C
C DETERMINE THE NUMBER OF THE EQUATION THAT IS TO BE MODIFIED.
C
              NEQN = (I-I1+1) + NX*((J-J1) + NY*(K-K1))
C
C TEMPORARY DATA HEADER TO DISPLAY RELEVANT PARAMETERS FOR DEBUGGING.
C
              IF (I1LEVL .GT. 1) CALL Q8EQHD(NEQN)
C
C CALCULATE THE PARAMETERS AT THE CURRENT GRID POINT THAT WILL BE USED
C FOR MODIFYING THE EQUATION COEFFICIENTS ON THE INTERFACE.
C
              CALL Q3PARM (I,J,K,X,DX,A,Y,DY,B,Z,DZ,C)
C
C MODIFY THE COEFFICIENTS OF THE EQUATIONS AT THE INTERFACE.
C
              RK  = RKB/RKA
              DX2 = DX**2.D0
              DY2 = DY**2.D0
              DZ2 = DZ**2.D0
C
C ********************  DETERMINE NEW COEFFICIENTS  ********************
C DETERMINE THE NEW VALUES OF THE COEFFICIENTS OF THE DISCRETE
C EQUATION AT THE POINT (X,Y,Z) WHICH LIES ON THE INTERFACE.
C
              CO(NEQN,CENTER) = (RK+C)/C + (1.D0+C)*DZ2/(A*DX2)
     A                                   + (1.D0+C)*DZ2/(B*DY2)
              CO(NEQN, LOWER) = -1.D0
              CO(NEQN, UPPER) = -RK/C
              CO(NEQN,  WEST) = -(1.D0+C)*DZ2/((1.D0+A)*DX2)
              CO(NEQN,  EAST) =  CO(NEQN,  WEST)/A
```

```
                CO(NEQN, SOUTH) = -(1.D0+C)*DZ2/((1.D0+B)*DY2)
                CO(NEQN, NORTH) =  CO(NEQN, SOUTH)/B
C
C   DETERMINE BASE VALUE FOR THE MODIFIED RIGHT HAND SIDE OF THE
C   DISCRETE EQUATION AT THE POINT (X,Y,Z) ON THE INTERFACE.
C
                CO(NEQN,   RHS) = (.5D0*DZ2/RKA)*(QA + C*QB)
C
C ********************  INVOKE BOUNDARY CONDITIONS  ********************
C MODIFY COEFFICIENTS IF THE GRID POINT IS ADJACENT TO A BOUNDARY.
C
                IF ((I .LE. 2) .OR. (I .GE. (I1NGRX-1)) .OR.
     A              (J .LE. 2) .OR. (J .GE. (I1NGRY-1)) .OR.
     B              (K .LE. 2) .OR. (K .GE. (I1NGRZ-1)))
     C          CALL Q3BNDY(I,J,K,NEQN,X,Y,Z,DX,DY,DZ,L3BNDY,CO,OLDCO)
C
C ******************  MODIFY  E L L P A C K  VARIABLES  ******************
C
                CALL Q3STCF(COEF,BBBB,IDCO,NEQN,L3CVCT,L3BNDY,MNEQ,MNCO,
     A                      CO)
C
C *************  TEMPORARY OUTPUT STATEMENTS FOR DEBUGGING ***************
C
      IF (I1LEVL .GT. 1)
     ACALL Q8EQDT(L3CVCT,NEQN,I,X,DX,A,NX,I1,IX1,IX2,J,Y,DY,B,NY,J1,
     B        JY1,JY2,K,Z,DZ,C,NZ,K1,   0,   0,COEF,IDCO,BBBB,MNEQ,MNCO)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE Q3PARM(I,J,K,X,DX,A,Y,DY,B,Z,DZ,C)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E  PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q3PARM RETURNS THE PARAMETERS NECESSARY TO MODIFY THE COEFFICIENTS
C   OF THE EQUATIONS ON THE INTERFACE OF A THREE DIMENSIONAL NON-
C   UNIFORM GRID.  THE VALUES RETURNED ARE:
C
C        X      -- X VALUE OF THE (I,J,K) GRID POINT.
C        Y      -- Y VALUE OF THE (I,J,K) GRID POINT.
C        Z      -- Z VALUE OF THE (I,J,K) GRID POINT.
C        DX     -- DISTANCE BETWEEN X(I,J,K) AND X(I-1,J,K).
C        DY     -- DISTANCE BETWEEN Y(I,J,K) AND Y(I,J-1,K).
C        DZ     -- DISTANCE BETWEEN Z(I,J,K) AND Z(I,J,K-1).
C        A     -- DISTANCE BETWEEN X(I,J,K) AND X(I+1,J,K) DIVIDED BY DX.
C        B     -- DISTANCE BETWEEN Y(I,J,K) AND X(I,J+1,K) DIVIDED BY DY.
C        C     -- DISTANCE BETWEEN Z(I,J,K) AND Z(I,J,K+1) DIVIDED BY DZ.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   FEBRUARY 1991
C
C-----------------------------------------------------------------------
C
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
      COMMON / C1GRDX / R1GRDX(1)
      COMMON / C1GRDY / R1GRDY(1)
      COMMON / C1GRDZ / R1GRDZ(1)
C
C SET X AXIS PARAMETERS THAT WILL BE USED TO MODIFY INTERFACE EQUATIONS.
C
                              X    = R1GRDX(I)
      IF (I .NE. 1)           XW   = R1GRDX(I-1)
      IF (I .NE. I1NGRX)      XE   = R1GRDX(I+1)
      IF (I .NE. 1) THEN
                              DX   = X-XW
                              A    = (XE-X)/DX
      ELSE
                              DX   = XE-X
                              A    = 1.D0
      ENDIF
      IF (I .EQ. I1NGRX)      A    = 1.D0
C
C SET Y AXIS PARAMETERS THAT WILL BE USED TO MODIFY INTERFACE EQUATIONS.
C
                              Y    = R1GRDY(J)
      IF (J .NE. 1)           YS   = R1GRDY(J-1)
      IF (J .NE. I1NGRY)      YN   = R1GRDY(J+1)
      IF (J .NE. 1) THEN
                              DY   = Y-YS
                              B    = (YN-Y)/DY
      ELSE
```

```
                              DY    = YN-Y
                              B     = 1.D0
            ENDIF
            IF (J .EQ. I1NGRY)  B     = 1.D0
      C
      C SET Z AXIS PARAMETERS THAT WILL BE USED TO MODIFY INTERFACE EQUATIONS.
      C
                              Z     = R1GRDZ(K)
            IF (K .NE. 1)       ZL    = R1GRDZ(K-1)
            IF (K .NE. I1NGRZ)  ZU    = R1GRDZ(K+1)
            IF (K .NE. 1) THEN
                              DZ    = Z-ZL
                              C     = (ZU-Z)/DZ
            ELSE
                              DZ    = ZU-Z
                              C     = 1.D0
            ENDIF
            IF(K .EQ. I1NGRZ)   C     = 1.D0
            RETURN
            END
```

```
      SUBROUTINE Q3BNDY(I,J,K,NEQN,X,Y,Z,DX,DY,DZ,L3BNDY,CO,OLDCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E  PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q3BNDY MODIFIES THE VALUES OF THE EQUATION COEFFICIENTS OF THE GRID
C   POINT (I,J,K) IS ON OR ADJACENT TO  A SPECIFIED BOUNDARY.
C
C   INPUTS FOR Q3BNDY ARE:
C
C        X,Y,Z -- X,Y,Z VALUE OF THE (I,J,K) GRID POINT.
C        DX    -- DISTANCE BETWEEN X(I,J,K) AND X(I-1,J,K).
C        DY    -- DISTANCE BETWEEN Y(I,J,K) AND Y(I,J-1,K).
C        DZ    -- DISTANCE BETWEEN Z(I,J,K) AND Z(I,J,K-1).
C        CO    -- TEMP ARRAY CONTAINING EQUATION COEFFICIENT VALUES.
C        NEQN  -- INDEX OF EQUATION CURRENTLY BEING MODIFIED.
C       OLDCO -- TEMP ARRAY CONTAINING PREVIOUS EQN. COEFFICIENT VALUES.
C       L3BNDY-- ARRAY OF LOGICAL ELEMENTS SPECIFYING IF A POINT LIES ON
C                A BOUNDARY OR IS OR ADJACENT TO A DIRICHLET BOUNDARY.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   FEBRUARY 1991
C
C-----------------------------------------------------------------------
C
      COMMON  / C1IVCN / I1LEVL, I1INPT, I1OUTP, I1SCRA, I1KWRK, I1KORD
      COMMON  / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
      COMMON  / C1RVGR / R1AXGR, R1AYGR, R1AZGR, R1BXGR, R1BYGR,
     A                   R1BZGR, R1HXGR, R1HYGR, R1HZGR
      COMMON  / C1BCTY / I1BCTY(1)
      DOUBLE PRECISION BVALUE(4)
      DOUBLE PRECISION CO(1:10000,8), OLDCO(1:10000,8)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS,ISDIRC
     A       /    1,    2,    3,    4,    5,    6,    7,   8,     1 /

      INTEGER BCOEUU, BCOEUX, BCOEUY, BCOEUZ
      DATA    BCOEUU, BCOEUX, BCOEUY, BCOEUZ
     A       /     1,      2,      3,      4 /
      LOGICAL L3BNDY(1:10000,6)
C
```

```
C ************************  EASTERN  BOUNDARY  ************************
C IF BOUNDARY MODIFICATIONS HAVE BEEN MADE BY A PREVIOUS CALL OF A
C CONVECTIVE SUBROUTINE MAKE NO FURTHER MODIFICATIONS ON EAST BOUNDARY.
C
      IF (L3BNDY(NEQN,EAST)) GOTO 48

C
C SET EAST COEFFICIENT.  MAKE MODIFICATIONS IF (X,Y,Z) IS
C ADJACENT TO THE EASTERN BOUNDARY.
C
      IF (I .GE. (I1NGRX-1)) BNDRHS = R1BRHS(EAST,R1BXGR,Y,Z)

      IF (I1LEVL .GT. 1) THEN
      IF((I.EQ.I1NGRX) .OR. (I.EQ.(I1NGRX-1) .AND. I1BCTY(EAST).EQ.1))
     A   CALL Q8BDAT(CO(NEQN,EAST), BNDRHS, I1BCTY(EAST),' EAST')
      ENDIF
C
C MODIFY RIGHT HAND SIDE OF EQUATION IF ADJACENT TO DIRICHLET BOUNDARY.
C
      IF (I .EQ. (I1NGRX-1) .AND. I1BCTY(EAST) .EQ. ISDIRC) THEN
           CO(NEQN,RHS) = CO(NEQN,RHS) - CO(NEQN,EAST)*BNDRHS
           L3BNDY(NEQN,EAST) = .TRUE.
           GOTO 50
      ENDIF
C
C MODIFY CENTER COEFFICIENT, WEST COEFFICIENT, AND RIGHT HAND SIDE
C OF EQUATION IF ADJACENT TO A MIXED BOUNDARY.
C
      IF (I .EQ. I1NGRX) THEN
           CALL Q1BCOE(EAST,R1BXGR,Y,Z,BVALUE)
           CU      = BVALUE(BCOEUU)
           CUX     = BVALUE(BCOEUX)
           CO(NEQN,CENTER) = CO(NEQN,CENTER)
     A                     - 2.D0*CU*CO(NEQN,EAST)*DX/CUX
           CO(NEQN,WEST)   = CO(NEQN,WEST) + CO(NEQN,EAST)
           CO(NEQN,RHS)    = CO(NEQN,RHS)
     A                     - 2.D0*CO(NEQN,EAST)*DX*BNDRHS/CUX
           L3BNDY(NEQN,EAST) = .TRUE.
      ENDIF
C
C IF THE CONVECTIVE SUBROUTINE SETS THE EASTERN COEFFICIENT EQUAL TO
C ZERO, THEN UNDO ON PREVIOUS BOUNDARY MODIFICATIONS THAT MAY HAVE
C BEEN MADE FOR AN EASTERN DIRICHLET BOUNDARY.
C
   48 IF (CO(NEQN,EAST) .EQ. 0.D0 .AND. I1BCTY(EAST) .EQ. ISDIRC
     A    .AND. OLDCO(NEQN,EAST) .NE. 0.D0 .AND. I.EQ.(I1NGRX-1))THEN
           CO(NEQN,RHS) = CO(NEQN,RHS) + OLDCO(NEQN,EAST)*BNDRHS
          IF (I1LEVL .GT. 1)
     A      CALL Q8BRST (OLDCO(NEQN,EAST),BNDRHS,I1BCTY(EAST),' EAST')
      ENDIF
C
C ************************  NORTHERN BOUNDARY  ************************
C IF BOUNDARY MODIFICATIONS HAVE BEEN MADE BY A PREVIOUS CALL OF A
C CONVECTIVE SUBROUTINE MAKE NO FURTHER MODIFICATIONS ON NORTH BOUNDARY.
C
   50 IF (L3BNDY(NEQN,NORTH)) GOTO 53
C
C SET NORTH COEFFICIENT.  MAKE MODIFICATIONS IF (X,Y,Z) IS
C ADJACENT TO THE NORTHERN BOUNDARY.
C
```

```
      IF (J .GE. (I1NGRY-1)) BNDRHS = R1BRHS(NORTH,X,R1BYGR,Z)
      IF (I1LEVL .GT. 1) THEN
      IF((J.EQ.I1NGRY) .OR. (J.EQ.(I1NGRY-1) .AND. I1BCTY(NORTH).EQ.1))
     A   CALL Q8BDAT(CO(NEQN,NORTH), BNDRHS, I1BCTY(NORTH),'NORTH')
      ENDIF
C
C MODIFY RIGHT HAND SIDE OF EQUATION IF ADJACENT TO DIRICHLET BOUNDARY
C
      IF (J .EQ. (I1NGRY-1) .AND. I1BCTY(NORTH) .EQ. ISDIRC) THEN
           CO(NEQN,RHS) = CO(NEQN,RHS) - CO(NEQN,NORTH)*BNDRHS
           L3BNDY(NEQN,NORTH) = .TRUE.
           GOTO 55
      ENDIF
C
C MODIFY CENTER COEFFICIENT, SOUTH COEFFICIENT, AND RIGHT HAND SIDE
C OF EQUATION IF ADJACENT TO A MIXED BOUNDARY.
C
      IF (J .EQ. I1NGRY) THEN
           CALL Q1BCOE(NORTH,X,R1BYGR,Z,BVALUE)
           CU     = BVALUE(BCOEUU)
           CUY    = BVALUE(BCOEUY)
           CO(NEQN,CENTER) = CO(NEQN,CENTER)
     A                     - 2.D0*CU*CO(NEQN,NORTH)*DY/CUY
           CO(NEQN,SOUTH)  = CO(NEQN,SOUTH) + CO(NEQN,NORTH)
           CO(NEQN,RHS)    = CO(NEQN,RHS)
     A                     - 2.D0*CO(NEQN,NORTH)*DY*BNDRHS/CUY
           L3BNDY(NEQN,NORTH)  = .TRUE.
           GOTO 55
      ENDIF
C
C IF THE CONVECTIVE SUBROUTINE SETS THE NORTHERN COEFFICIENT EQUAL TO
C ZERO, THEN UNDO ON PREVIOUS BOUNDARY MODIFICATIONS THAT MAY HAVE
C BEEN MADE FOR A NORTHERN DIRICHLET BOUNDARY.
C
   53 IF (CO(NEQN,NORTH) .EQ. 0.D0 .AND. I1BCTY(NORTH) .EQ. ISDIRC
     A    .AND. OLDCO(NEQN,NORTH) .NE. 0.D0 .AND. J.EQ.(I1NGRY-1))THEN
           CO(NEQN,RHS) = CO(NEQN,RHS) + OLDCO(NEQN,NORTH)*BNDRHS
          IF (I1LEVL .GT. 1)
     A      CALL Q8BRST (OLDCO(NEQN,NORTH),BNDRHS,I1BCTY(NORTH),'NORTH')
      ENDIF
C
```

```
C ************************ UPPER    BOUNDARY ****************************
C IF BOUNDARY MODIFICATIONS HAVE BEEN MADE BY A PREVIOUS CALL OF A
C CONVECTIVE SUBROUTINE MAKE NO FURTHER MODIFICATIONS ON UPPER BOUNDARY.
C
   55 IF (L3BNDY(NEQN,UPPER)) GOTO 58
C
C SET UPPER COEFFICIENT.  MAKE MODIFICATIONS IF (X,Y,Z) IS
C ADJACENT TO THE UPPER BOUNDARY.
C
      IF (K .GE. (I1NGRZ-1)) BNDRHS = R1BRHS(UPPER,X,Y,R1BZGR)
      IF (I1LEVL .GT. 1)THEN
      IF((K.EQ.I1NGRZ) .OR. (K.EQ.(I1NGRZ-1) .AND. I1BCTY(UPPER).EQ.1))
     A   CALL Q8BDAT(CO(NEQN,UPPER), BNDRHS, I1BCTY(UPPER),'UPPER')
      ENDIF
C
C MODIFY RIGHT HAND SIDE OF EQUATION IF ADJACENT TO DIRICHLET BOUNDARY.
C
      IF (K .EQ. (I1NGRZ-1) .AND. I1BCTY(UPPER) .EQ. ISDIRC) THEN
           CO(NEQN,RHS)    = CO(NEQN,RHS) - CO(NEQN,UPPER)*BNDRHS
           L3BNDY(NEQN,UPPER) = .TRUE.
           GOTO 60
      ENDIF
C
C MODIFY CENTER COEFFICIENT, LOWER COEFFICIENT, AND RIGHT HAND SIDE
C OF EQUATION IF ADJACENT TO A MIXED BOUNDARY.
C
      IF (K .EQ. I1NGRZ) THEN
           CALL Q1BCOE(UPPER,X,Y,R1BZGR,BVALUE)
           CU      = BVALUE(BCOEUU)
           CUZ     = BVALUE(BCOEUZ)
           CO(NEQN,CENTER) = CO(NEQN,CENTER)
     A                     - 2.D0*CU*CO(NEQN,UPPER)*DZ/CUZ
           CO(NEQN,LOWER) = CO(NEQN,LOWER) + CO(NEQN,UPPER)
           CO(NEQN,RHS)    = CO(NEQN,RHS)
     A                     - 2.D0*CO(NEQN,UPPER)*DZ*BNDRHS/CUZ
           L3BNDY(NEQN,UPPER) = .TRUE.
      ENDIF
C
C IF THE CONVECTIVE SUBROUTINE SETS THE UPPER COEFFICIENT EQUAL TO
C ZERO, THEN UNDO ON PREVIOUS BOUNDARY MODIFICATIONS THAT MAY HAVE
C BEEN MADE FOR AN UPPER DIRICHLET BOUNDARY.
C
   58 IF (CO(NEQN,UPPER) .EQ. 0.D0 .AND. I1BCTY(UPPER) .EQ. ISDIRC
     A     .AND. OLDCO(NEQN,UPPER) .NE. 0.D0 .AND. K.EQ.(I1NGRZ-1))THEN
           CO(NEQN,RHS) = CO(NEQN,RHS) + OLDCO(NEQN,UPPER)*BNDRHS
         IF (I1LEVL .GT. 1)
     A      CALL Q8BRST (OLDCO(NEQN,UPPER),BNDRHS,I1BCTY(UPPER),'UPPER')
      ENDIF
C
```

```
C ************************ WESTERN BOUNDARY ****************************
C IF BOUNDARY MODIFICATIONS HAVE BEEN MADE BY A PREVIOUS CALL OF A CON-
C VECTIVE SUBROUTINE MAKE NO FURTHER MODIFICATIONS ON WESTERN BOUNDARY.
C
   60 IF (L3BNDY(NEQN,WEST)) GOTO 63
C
C SET WEST COEFFICIENT.  MAKE MODIFICATIONS IF (X,Y,Z) IS
C ADJACENT TO THE WESTERN BOUNDARY.
C
      IF (I .LE. 2) BNDRHS = R1BRHS(WEST,R1AXGR,Y,Z)
      IF (I1LEVL .GT. 1) THEN
      IF((I .EQ. 1) .OR. (I. EQ. 2 .AND. I1BCTY(WEST) .EQ. ISDIRC))
     A   CALL Q8BDAT(CO(NEQN,WEST), BNDRHS, I1BCTY(WEST),' WEST')
      ENDIF
C
C MODIFY RIGHT HAND SIDE OF EQUATION IF ADJACENT TO DIRICHLET BOUNDARY.
C
      IF (I .EQ. 2 .AND. I1BCTY(WEST) .EQ. ISDIRC) THEN
           CO(NEQN,RHS)  = CO(NEQN,RHS) - CO(NEQN,WEST)*BNDRHS
           L3BNDY(NEQN,WEST) = .TRUE.
           GOTO 65
      ENDIF
C
C MODIFY CENTER COEFFICIENT, EAST COEFFICIENT, AND RIGHT HAND SIDE
C OF EQUATION IF ADJACENT TO A MIXED BOUNDARY.
C
      IF (I .EQ. 1) THEN
           CALL Q1BCOE(WEST,R1AXGR,Y,Z,BVALUE)
           CU      = BVALUE(BCOEUU)
           CUX     = BVALUE(BCOEUX)
           CO(NEQN,CENTER) = CO(NEQN,CENTER)
     A                       - 2.D0*CU*CO(NEQN,WEST)*DX/CUX
           CO(NEQN,EAST)   = CO(NEQN,EAST) + CO(NEQN,WEST)
           CO(NEQN,RHS)    = CO(NEQN,RHS)
     A                      - 2.D0*CO(NEQN,WEST)*DX*BNDRHS/CUX
           L3BNDY(NEQN,WEST) = .TRUE.
      ENDIF
C
C IF THE CONVECTIVE SUBROUTINE SETS THE WESTERN COEFFICIENT EQUAL TO
C ZERO, THEN UNDO ON PREVIOUS BOUNDARY MODIFICATIONS THAT MAY HAVE
C BEEN MADE FOR A WESTERN DIRICHLET BOUNDARY.
C
   63 IF (CO(NEQN,WEST) .EQ. 0.D0 .AND. I1BCTY(WEST) .EQ. ISDIRC
     A     .AND. OLDCO(NEQN,WEST) .NE. 0.D0 .AND. I .EQ. 2)THEN
           CO(NEQN,RHS) = CO(NEQN,RHS) + OLDCO(NEQN,WEST)*BNDRHS
         IF (I1LEVL .GT. 1)
     A      CALL Q8BRST (OLDCO(NEQN,WEST),BNDRHS,I1BCTY(WEST),' WEST')
      ENDIF
C
```

```
C ************************  SOUTHERN BOUNDARY  ****************************
C IF BOUNDARY MODIFICATIONS HAVE BEEN MADE BY A PREVIOUS CALL OF A CON-
C VECTIVE SUBROUTINE MAKE NO FURTHER MODIFICATIONS ON SOUTHERN BOUNDARY.
C
   65 IF (L3BNDY(NEQN,SOUTH)) GOTO 68
C
C SET SOUTH COEFFICIENT.  MAKE MODIFICATIONS IF (X,Y,Z) IS
C ADJACENT TO THE SOUTHERN BOUNDARY.
C
      IF (J .LE. 2)  BNDRHS = R1BRHS(SOUTH,X,R1AYGR,Z)
      IF (I1LEVL .GT. 1) THEN
      IF((J .EQ. 1) .OR. (J .EQ. 2  .AND. I1BCTY(SOUTH) .EQ. ISDIRC))
     A   CALL Q8BDAT(CO(NEQN,SOUTH), BNDRHS, I1BCTY(SOUTH),'SOUTH')
      ENDIF
C
C MODIFY RIGHT HAND SIDE OF EQUATION IF ADJACENT TO DIRICHLET BOUNDARY.
C
      IF (J .EQ. 2 .AND. I1BCTY(SOUTH) .EQ. ISDIRC) THEN
           CO(NEQN,RHS)    = CO(NEQN,RHS) - CO(NEQN,SOUTH)*BNDRHS
           L3BNDY(NEQN,SOUTH)  = .TRUE.
           GOTO 70
      ENDIF
C
C MODIFY CENTER COEFFICIENT, NORTH COEFFICIENT, AND RIGHT HAND SIDE
C OF EQUATION IF ADJACENT TO A MIXED BOUNDARY.
C
      IF (J .EQ. 1) THEN
           CALL Q1BCOE(SOUTH,X,R1AYGR,Z,BVALUE)
           CU      = BVALUE(BCOEUU)
           CUY     = BVALUE(BCOEUY)
           CO(NEQN,CENTER) = CO(NEQN,CENTER)
     A                     - 2.D0*CU*CO(NEQN,SOUTH)*DY/CUY
           CO(NEQN,NORTH)  = CO(NEQN,NORTH) + CO(NEQN,SOUTH)
           CO(NEQN,RHS)    = CO(NEQN,RHS)
     A                    - 2.D0*CO(NEQN,SOUTH)*DY*BNDRHS/CUY
           L3BNDY(NEQN,SOUTH) = .TRUE.
      ENDIF
C
C IF THE CONVECTIVE SUBROUTINE SETS THE SOUTHERN COEFFICIENT EQUAL TO
C ZERO, THEN UNDO ON PREVIOUS BOUNDARY MODIFICATIONS THAT MAY HAVE
C BEEN MADE FOR A SOUTHERN DIRICHLET BOUNDARY.
C
   68 IF (CO(NEQN,SOUTH) .EQ. 0.D0 .AND. I1BCTY(SOUTH) .EQ. ISDIRC
     A    .AND. OLDCO(NEQN,SOUTH) .NE. 0.D0 .AND. J .EQ. 2) THEN
           CO(NEQN,RHS) = CO(NEQN,RHS) + OLDCO(NEQN,SOUTH)*BNDRHS
         IF (I1LEVL .GT. 1)
     A      CALL Q8BRST (OLDCO(NEQN,SOUTH),BNDRHS,I1BCTY(SOUTH),'SOUTH')
      ENDIF
C
```

```
C ************************ LOWER    BOUNDARY ****************************
C IF BOUNDARY MODIFICATIONS HAVE BEEN MADE BY A PREVIOUS CALL OF A
C CONVECTIVE SUBROUTINE MAKE NO FURTHER MODIFICATIONS ON LOWER BOUNDARY.
C
   70 IF (L3BNDY(NEQN,LOWER)) GOTO 73
C
C SET LOWER COEFFICIENT.  MAKE MODIFICATIONS IF (X,Y,Z) IS
C ADJACENT TO THE LOWER BOUNDARY.
C
      IF (K .LE. 2) BNDRHS = R1BRHS(LOWER,X,Y,R1AZGR)
      IF (I1LEVL .GT. 1) THEN
      IF((K .EQ. 1) .OR. (K .EQ. 2 .AND. I1BCTY(LOWER) .EQ. ISDIRC))
     A   CALL Q8BDAT(CO(NEQN,LOWER), BNDRHS, I1BCTY(LOWER),'LOWER')
      ENDIF
C
C MODIFY RIGHT HAND SIDE OF EQUATION IF ADJACENT TO DIRICHLET BOUNDARY.
C
      IF (K .EQ. 2 .AND. I1BCTY(LOWER) .EQ. ISDIRC) THEN
           CO(NEQN,RHS)    = CO(NEQN,RHS) - CO(NEQN,LOWER)*BNDRHS
           L3BNDY(NEQN,LOWER) = .TRUE.
           GOTO 75
      ENDIF
C
C MODIFY CENTER COEFFICIENT, UPPER COEFFICIENT, AND RIGHT HAND SIDE
C OF EQUATION IF ADJACENT TO A MIXED BOUNDARY.
C
      IF (K .EQ. 1) THEN
           CALL Q1BCOE(LOWER,X,Y,R1AZGR,BVALUE)
           CU      = BVALUE(BCOEUU)
           CUZ     = BVALUE(BCOEUZ)
           CO(NEQN,CENTER) = CO(NEQN,CENTER)
     A                     - 2.D0*CU*CO(NEQN,LOWER)*DZ/CUZ
           CO(NEQN,UPPER) = CO(NEQN,UPPER) + CO(NEQN,LOWER)
           CO(NEQN,RHS)    = CO(NEQN,RHS)
     A                     - 2.D0*CO(NEQN,LOWER)*DZ*BNDRHS/CUZ
           L3BNDY(NEQN,LOWER) = .TRUE.
      ENDIF
C
C IF THE CONVECTIVE SUBROUTINE SETS THE LOWER COEFFICIENT EQUAL TO
C ZERO, THEN UNDO ON PREVIOUS BOUNDARY MODIFICATIONS THAT MAY HAVE
C BEEN MADE FOR A LOWER DIRICHLET BOUNDARY.
C
   73 IF (CO(NEQN,LOWER) .EQ. 0.D0 .AND. I1BCTY(LOWER) .EQ. ISDIRC
     A    .AND. OLDCO(NEQN,LOWER) .NE. 0.D0 .AND. K .EQ. 2) THEN
           CO(NEQN,RHS) = CO(NEQN,RHS) + OLDCO(NEQN,LOWER)*BNDRHS
         IF (I1LEVL .GT. 1)
     A      CALL Q8BRST (OLDCO(NEQN,LOWER),BNDRHS,I1BCTY(LOWER),'LOWER')
      ENDIF
   75 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE Q3STCF(COEF,BBBB,IDCO,NEQN,L3CVCT,L3BNDY,MNEQ,MNCO,
     A                  CO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E  PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q3STCF MODIFIES THE  E L L P A C K  VARIABLES THAT DESCRIBE THE
C   EQUATION ALONG THE INTERFACE OF THE THREE DIMENSIONAL REGIME.
C   INPUTS FOR Q3STCF INCLUDE:
C
C    CO    -- TEMP. ARRAY CONTAINING EQUATION COEFFICIENT VALUES.
C    NEQN  -- INDEX OF EQUATION CURRENTLY BEING MODIFIED.
C    L3CVCT-- ARRAY OF LOGICAL ELEMENTS SPECIFYING IF A POINT LIES
C             ON ONE OR MORE CONVECTIVE INTERFACES.
C    MNEQ  -- NUMBER OF EQUATIONS IN THE SYSTEM.
C    MNCO  -- NUMBER OF COEFFICIENTS PER EQUATION.
C    COEF  -- COEFFICIENT FOR EACH VARIABLE OF EACH EQUATION.
C    BBBB  -- RIGHT HAND SIDE VALUE OF EACH EQUATION.
C    IDCO  -- INTEGER SUBSCRIPT THAT IDENTIFIES POSITION OF THE
C             VARIABLE WITHIN THE GRID.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   FEBRUARY 1991
C
C  N O T E
C
C   USES JUNE 1978/SEPTEMBER 1982 VERSION OF 7 POINT 3D M O D U L E
C   WRITTEN BY ROGER GRIMES.
C
C-----------------------------------------------------------------------
C
      INTEGER IDCO(MNEQ, MNCO)
      DOUBLE  PRECISION  COEF(MNEQ, MNCO)
      DOUBLE  PRECISION  BBBB(MNEQ)
      DOUBLE  PRECISION  CO(1:10000,8)
      LOGICAL L3CVCT(1:10000,6), L3BNDY(1:10000,6)
      INTEGER EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
      DATA    EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS,ISDIRC
     A       /   1,    2,   3,    4,    5,    6,     7,  8,     1 /
C
C SET CENTER COEFFICIENT.  ALL OTHER COEFFICIENTS WILL BE NORMALIZED
C WITH RESPECT TO THE CENTER COEFFICIENT.
C
      ICOL = 1
      COEF(NEQN,ICOL) = CO(NEQN,CENTER)
C
C SET EAST COEFFICIENT IF (X,Y,Z) IS AN INTERIOR POINT.
C
      IF (L3CVCT(NEQN, EAST) .OR. L3BNDY(NEQN,EAST)) GOTO 5
           ICOL = ICOL + 1
           COEF(NEQN,ICOL) = CO(NEQN,EAST)
C
```

```
C SET NORTH COEFFICIENT IF (X,Y,Z) IS AN INTERIOR POINT.
C
    5 IF (L3CVCT(NEQN,NORTH) .OR. L3BNDY(NEQN,NORTH)) GOTO 10
        ICOL = ICOL + 1
        COEF(NEQN,ICOL) = CO(NEQN,NORTH)
C
C SET UPPER COEFFICIENT IF (X,Y,Z) IS AN INTERIOR POINT.
C
   10 IF (L3CVCT(NEQN,UPPER) .OR. L3BNDY(NEQN,UPPER)) GOTO 15
        ICOL = ICOL + 1
        COEF(NEQN,ICOL) = CO(NEQN,UPPER)
C
C SET WEST COEFFICIENT IF (X,Y,Z) IS AN INTERIOR POINT.
C
   15 IF (L3CVCT(NEQN, WEST) .OR. L3BNDY(NEQN,WEST)) GOTO 20
        ICOL = ICOL + 1
        COEF(NEQN,ICOL) = CO(NEQN,WEST)
C
C SET SOUTH COEFFICIENT IF (X,Y,Z) IS AN INTERIOR POINT.
C
   20 IF (L3CVCT(NEQN,SOUTH) .OR. L3BNDY(NEQN,SOUTH)) GOTO 25
        ICOL = ICOL + 1
        COEF(NEQN,ICOL) = CO(NEQN,SOUTH)
C
C SET LOWER COEFFICIENT IF (X,Y,Z) IS AN INTERIOR POINT.
C
   25 IF (L3CVCT(NEQN,LOWER) .OR. L3BNDY(NEQN,LOWER)) GOTO 30
        ICOL = ICOL + 1
        COEF(NEQN,ICOL) = CO(NEQN,LOWER)
C
C SET VALUE FOR RIGHT SIDE EQUATION AT (X,Y,Z).
C
   30 BBBB(NEQN)    = CO(NEQN,RHS)
      RETURN
      END
```

```
      SUBROUTINE Q4XRNG(X,Y1,Y2,Z1,Z2,IX,JY1,JY2,KZ1,KZ2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q4XRNG DETERMINES THE RANGE OF THE UNKNOWN GRID POINTS ALONG AN
C   X INTERFACE.  IT ENSURES THAT THE BOUNDS SPECIFIED FOR THE X
C   INTERFACE HAVE ALSO BEEN SPECIFIED ON THE GRID.  IF THEY HAVE NOT
C   THEN THE RUN IS TERMINATED.  INPUTS ARE:
C
C        X        -- LOCATION OF THE X INTERFACE.
C        Y1, Y2   -- Y BOUNDS OF THE X INTERFACE.
C        Z1, Z1   -- Z BOUNDS OF THE X INTERFACE.
C
C   VALUES RETURNED ARE:
C
C        IX       -- X GRID POSITION  OF THE X INTERFACE.
C        JY1, JY2 -- Y GRID POSITIONS OF THE Y BOUNDS.
C        KZ1, KZ2 -- Z GRID POSITIONS OF THE Z BOUNDS.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   FEBRUARY 1991
C
C-----------------------------------------------------------------------
C
      COMMON / C1BCTY / I1BCTY(1)
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
      COMMON / C1GRDX / R1GRDX(1)
      COMMON / C1GRDY / R1GRDY(1)
      COMMON / C1GRDZ / R1GRDZ(1)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,ISDIRC
     A       /    1,    2,    3,    4,    5,    6,     1 /
C
C ENSURE THAT INTERFACE PARAMETERS HAVE BEEN SPECIFIED ON THE GRID.
C
      IX = 0
      DO 10 I = 1, I1NGRX
           IF (DABS(R1GRDX(I)-X) .LE. 1E-6) IX = I
   10 CONTINUE
C
C TERMINATE PROGRAM IF INTERFACE WAS NOT PROPERLY SPECIFIED.
C
      IF (IX .EQ. 0) THEN
           CALL Q1ERRH(17HSUBROUTINE Q4XRNG,17)
           WRITE (6,15)
   15      FORMAT(/'X INTERFACE POSITION MUST BE SPECIFIED ON GRID'/)
           CALL Q1ERRT
           RETURN
      ENDIF
      IF (IX .EQ. 1 .OR. IX .EQ. I1NGRX) THEN
           CALL Q1ERRH(17HSUBROUTINE Q4XRNG,17)
```

```
            WRITE (6,25)
   25       FORMAT(/'X INTERFACE CAN NOT EXIST ON SPECIFIED BOUNDARY'/)
            CALL Q1ERRT
            RETURN
         ENDIF
C
C TERMINATE PROGRAM IF Y BOUNDS OF INTERFACE ARE NOT SPECIFIED ON GRID.
C
         JY1 = 0
         JY2 = 0
         DO 30 J = 1, I1NGRY
            IF (DABS(R1GRDY(J)-Y1) .LE. 1E-6) JY1 = J
            IF (DABS(R1GRDY(J)-Y2) .LE. 1E-6) JY2 = J
   30    CONTINUE
         IF (JY1 .EQ. 0 .OR. JY2 .EQ. 0) THEN
            CALL Q1ERRH(17HSUBROUTINE Q4XRNG,17)
            WRITE (6,35)
   35       FORMAT(/'X INTERFACE Y BOUNDS MUST BE SPECIFIED ON GRID'/)

            CALL Q1ERRT
            RETURN
         ENDIF
C
C MODIFY JY1 AND JY2 IF Y1 AND Y2 ARE ON DIRICHLET BOUNDARIES.
C
         IF (JY1 .EQ. 1       .AND. I1BCTY(SOUTH) .EQ. ISDIRC) JY1 = 2
         IF (JY2 .EQ. I1NGRY .AND. I1BCTY(NORTH) .EQ. ISDIRC) JY2 = JY2-1
C
C TERMINATE PROGRAM IF Z BOUNDS OF INTERFACE ARE NOT SPECIFIED ON GRID.
C
         KZ1 = 0
         KZ2 = 0
         DO 40 K = 1, I1NGRZ
            IF (DABS(R1GRDZ(K)-Z1) .LE. 1E-6) KZ1 = K
            IF (DABS(R1GRDZ(K)-Z2) .LE. 1E-6) KZ2 = K
   40    CONTINUE
         IF (KZ1 .EQ. 0 .OR. KZ2 .EQ. 0) THEN
            CALL Q1ERRH(17HSUBROUTINE Q4XRNG,17)
            WRITE (6,45)
   45       FORMAT(/'X INTERFACE Z BOUNDS MUST BE SPECIFIED ON GRID'/)
            CALL Q1ERRT
            RETURN
         ENDIF
C
C MODIFY KZ1 AND KZ2 IF Z1 AND Z2 ARE ON DIRICHLET BOUNDARIES.
C
         IF (KZ1 .EQ. 1       .AND. I1BCTY(LOWER) .EQ. ISDIRC) KZ1 = 2
         IF (KZ2 .EQ. I1NGRZ .AND. I1BCTY(UPPER) .EQ. ISDIRC) KZ2 = KZ2-1
         RETURN
         END
```

```
      SUBROUTINE Q4YRNG(Y,X1,X2,Z1,Z2,JY,IX1,IX2,KZ1,KZ2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q4YRNG DETERMINES THE RANGE OF THE UNKNOWN GRID POINTS ALONG AN
C   Y INTERFACE.  IT ENSURES THAT THE BOUNDS SPECIFIED FOR THE Y
C   INTERFACE HAVE ALSO BEEN SPECIFIED ON THE GRID.  IF THEY HAVE NOT
C   THEN THE RUN IS TERMINATED.  INPUTS ARE:
C
C        X1, X2   -- X BOUNDS OF THE Y INTERFACE.
C        Y        -- LOCATION OF THE Y INTERFACE.
C        Z1, Z1   -- Z BOUNDS OF THE Y INTERFACE.
C
C   VALUES RETURNED ARE:
C
C        IX1, IX2 -- X GRID POSITIONS OF THE X BOUNDS.
C        JY       -- Y GRID POSITION  OF THE Y INTERFACE.
C        KZ1, KZ2 -- Z GRID POSITIONS OF THE Z BOUNDS.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   MARCH 1991
C
C-----------------------------------------------------------------------
C
      COMMON / C1BCTY / I1BCTY(1)
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
      COMMON / C1GRDX / R1GRDX(1)
      COMMON / C1GRDY / R1GRDY(1)
      COMMON / C1GRDZ / R1GRDZ(1)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,ISDIRC
     A       /    1,    2,    3,    4,    5,    6,     1 /
C
C ENSURE THAT INTERFACE PARAMETERS HAVE BEEN SPECIFIED ON THE GRID
C
      JY = 0
      DO 10 J = 1, I1NGRY
           IF (DABS(R1GRDY(J)-Y) .LE. 1E-6) JY = J
   10 CONTINUE
C
C TERMINATE PROGRAM IF INTERFACE WAS NOT PROPERLY SPECIFIED.
C
      IF (JY .EQ. 0) THEN
           CALL Q1ERRH(17HSUBROUTINE Q4YRNG,17)
           WRITE (6,15)
   15      FORMAT(/'Y INTERFACE POSITION MUST BE SPECIFIED ON GRID'/)
           CALL Q1ERRT
           RETURN
      ENDIF
      IF (JY .EQ. 1 .OR. JY .EQ. I1NGRY) THEN
           CALL Q1ERRH(17HSUBROUTINE Q4YRNG,17)
```

```
            WRITE (6,25)
   25       FORMAT(/'Y INTERFACE CAN NOT EXIST ON A SPECIFIED BOUNDARY'/)
            CALL Q1ERRT
            RETURN
         ENDIF
C
C TERMINATE PROGRAM IF X BOUNDS OF INTERFACE ARE NOT SPECIFIED ON GRID.
C
         IX1 = 0
         IX2 = 0
         DO 30 I = 1, I1NGRX
            IF (DABS(R1GRDX(I)-X1) .LE. 1E-6) IX1 = I
            IF (DABS(R1GRDX(I)-X2) .LE. 1E-6) IX2 = I
   30    CONTINUE
         IF (IX1 .EQ. 0 .OR. IX2 .EQ. 0) THEN
            CALL Q1ERRH(17HSUBROUTINE Q4YRNG,17)
            WRITE (6,35)
   35       FORMAT(/'Y INTERFACE X BOUNDS MUST BE SPECIFIED ON GRID'/)

            CALL Q1ERRT
            RETURN
         ENDIF
C
C MODIFY IX1 AND IX2 IF X1 AND X2 ARE ON DIRICHLET BOUNDARIES.
C
         IF (IX1 .EQ. 1       .AND. I1BCTY(WEST) .EQ. ISDIRC) IX1 = 2
         IF (IX2 .EQ. I1NGRX .AND. I1BCTY(EAST) .EQ. ISDIRC) IX2 = IX2-1
C
C TERMINATE PROGRAM IF Z BOUNDS OF INTERFACE ARE NOT SPECIFIED ON GRID.
C
         KZ1 = 0
         KZ2 = 0
         DO 40 K = 1, I1NGRZ
            IF (DABS(R1GRDZ(K)-Z1) .LE. 1E-6) KZ1 = K
            IF (DABS(R1GRDZ(K)-Z2) .LE. 1E-6) KZ2 = K
   40    CONTINUE
         IF (KZ1 .EQ. 0 .OR. KZ2 .EQ. 0) THEN
            CALL Q1ERRH(17HSUBROUTINE Q4YRNG,17)
            WRITE (6,45)
   45       FORMAT(/'Y INTERFACE Z BOUNDS MUST BE SPECIFIED ON GRID'/)
            CALL Q1ERRT
            RETURN
         ENDIF
C
C MODIFY KZ1 AND KZ2 IF Z1 AND Z2 ARE ON DIRICHLET BOUNDARIES.
C
         IF (KZ1 .EQ. 1       .AND. I1BCTY(LOWER) .EQ. ISDIRC) KZ1 = 2
         IF (KZ2 .EQ. I1NGRZ .AND. I1BCTY(UPPER) .EQ. ISDIRC) KZ2 = KZ2-1
         RETURN
         END
```

```
      SUBROUTINE Q4ZRNG(Z,X1,X2,Y1,Y2,KZ,IX1,IX2,JY1,JY2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  DISCRETIZATION  M O D U L E PHASE 3D
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q4ZRNG DETERMINES THE RANGE OF THE UNKNOWN GRID POINTS ALONG AN
C   Z INTERFACE.  IT ENSURES THAT THE BOUNDS SPECIFIED FOR THE Z
C   INTERFACE HAVE ALSO BEEN SPECIFIED ON THE GRID.  IF THEY HAVE NOT
C   THEN THE RUN IS TERMINATED.  INPUTS ARE:
C
C        X1, X2   -- X BOUNDS OF THE Z INTERFACE.
C        Y1, Y2   -- Y BOUNDS OF THE Z INTERFACE.
C        Z        -- LOCATION OF THE Z INTERFACE.
C
C   VALUES RETURNED ARE:
C
C        IX1, IX2 -- X GRID POSITIONS OF THE X BOUNDS.
C        JY1, JY2 -- Y GRID POSITION  OF THE Y BOUNDS.
C        KZ       -- Z GRID POSITION  OF THE Z INTERFACE.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   MARCH 1991
C
C-----------------------------------------------------------------------
C
      COMMON / C1BCTY / I1BCTY(1)
      COMMON / C1IVGR / I1NGRX, I1NGRY, I1NGRZ, I1NBPT, I1MBPT, I1PACK
      COMMON / C1GRDX / R1GRDX(1)
      COMMON / C1GRDY / R1GRDY(1)
      COMMON / C1GRDZ / R1GRDZ(1)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,ISDIRC
     A       /    1,    2,    3,    4,    5,    6,     1 /
C
C ENSURE THAT INTERFACE PARAMETERS HAVE BEEN SPECIFIED ON THE GRID.
C
      KZ = 0
      DO 10 K = 1, I1NGRZ
           IF (DABS(R1GRDZ(K)-Z) .LE. 1E-6) KZ = K
   10 CONTINUE
C
C TERMINATE PROGRAM IF INTERFACE WAS NOT PROPERLY SPECIFIED.
C
      IF (KZ .EQ. 0) THEN
           CALL Q1ERRH(17HSUBROUTINE Q4ZRNG,17)
           WRITE (6,15)
   15      FORMAT(/'Z INTERFACE POSITION MUST BE SPECIFIED ON GRID'/)
           CALL Q1ERRT
           RETURN
      ENDIF
      IF (KZ .EQ. 1 .OR. KZ .EQ. I1NGRZ) THEN
           CALL Q1ERRH(17HSUBROUTINE Q4ZRNG,17)
```

```
            WRITE (6,25)
   25       FORMAT('Z INTERFACE CAN NOT EXIST ON A SPECIFIED BOUNDARY')
            CALL Q1ERRT
            RETURN
         ENDIF
C
C TERMINATE PROGRAM IF X BOUNDS OF INTERFACE ARE NOT SPECIFIED ON GRID.
C
         IX1 = 0
         IX2 = 0
         DO 30 I = 1, I1NGRX
            IF (DABS(R1GRDX(I)-X1) .LE. 1E-6) IX1 = I
            IF (DABS(R1GRDX(I)-X2) .LE. 1E-6) IX2 = I
   30    CONTINUE
         IF (IX1 .EQ. 0 .OR. IX2 .EQ. 0) THEN
            CALL Q1ERRH(17HSUBROUTINE Q4ZRNG,17)
            WRITE (6,35)
   35       FORMAT('Z INTERFACE X BOUNDS MUST BE SPECIFIED ON GRID')

            CALL Q1ERRT
            RETURN
         ENDIF
C
C MODIFY IX1 AND IX2 IF X1 AND X2 ARE ON DIRICHLET BOUNDARIES.
C
         IF (IX1 .EQ. 1       .AND. I1BCTY(WEST) .EQ. ISDIRC) IX1 = 2
         IF (IX2 .EQ. I1NGRX .AND. I1BCTY(EAST) .EQ. ISDIRC) IX2 = IX2-1
C
C TERMINATE PROGRAM IF Y BOUNDS OF INTERFACE ARE NOT SPECIFIED ON GRID.
C
         JY1 = 0
         JY2 = 0
         DO 40 J = 1, I1NGRY
            IF (DABS(R1GRDY(J)-Y1) .LE. 1E-6) JY1 = J
            IF (DABS(R1GRDY(J)-Y2) .LE. 1E-6) JY2 = J
   40    CONTINUE
         IF (JY1 .EQ. 0 .OR. JY2 .EQ. 0) THEN
            CALL Q1ERRH(17HSUBROUTINE Q4ZRNG,17)
            WRITE (6,45)
   45       FORMAT('Z INTERFACE Y BOUNDS MUST BE SPECIFIED ON GRID')
            CALL Q1ERRT
            RETURN
         ENDIF
C
C MODIFY JY1 AND JY2 IF Y1 AND Y2 ARE ON DIRICHLET BOUNDARIES.
C
         IF (JY1 .EQ. 1       .AND. I1BCTY(SOUTH) .EQ. ISDIRC) JY1 = 2
         IF (JY2 .EQ. I1NGRY .AND. I1BCTY(NORTH) .EQ. ISDIRC) JY2 = JY2-1
         RETURN
         END
```

```
      SUBROUTINE Q5SORM(BBBB,COEF,IDCO,OMEGA,ZETA,ITMAX,UNKN,MNEQ,MNCO)
      IMPLICIT DOUBLE PRECISION (A-H,O-P)
C
C-----------------------------------------------------------------------
C  E L L P A C K  SOLUTION  M O D U L E  SOR
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q5SORM SOLVES THE SYSTEM OF EQUATIONS GENERATED BY THE DISCRETIZATION
C   ROUTINES USING STANDARD OVER-RELAXATION ITERATIVE METHOD.  INPUTS FOR
C   Q5SORM INCLUDE:
C
C    OMEGA -- RELAXATION FACTOR.
C    ZETA  -- CONVERGENCE TOLERANCE SETTING.
C    ITMAX -- MAXIMUM ITERATIONS PERFORMED BEFORE ROUTINE TERMINATES.
C             ON ONE OR MORE CONVECTIVE INTERFACES.
C    MNEQ  -- NUMBER OF EQUATIONS IN THE SYSTEM.
C    MNCO  -- NUMBER OF COEFFICIENTS PER EQUATION.
C    COEF  -- COEFFICIENT FOR EACH VARIABLE OF EACH EQUATION.
C    BBBB  -- RIGHT HAND SIDE VALUE OF EACH EQUATION.
C    IDCO  -- INTEGER SUBSCRIPT THAT IDENTIFIES POSITION OF THE
C             VARIABLE WITHIN THE GRID.
C    UNKN  -- VECTOR TO CONTAIN VALUES OF UNKNOWN VARIABLES.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   MAY 1991
C
C-----------------------------------------------------------------------
C
      COMMON  / C1IVCN / LEVEL, I1INPT, I1OUTP, I1SCRA, I1KWRK, I1KORD
      DOUBLE PRECISION COEF(MNEQ,MNCO), BBBB(MNEQ), UNKN(1:10000)
      INTEGER IDCO(MNEQ,MNCO), N(9), NEQN(9)
C
C PRINT SOR HEADER OF LEVEL > 0.
C
      IF (LEVEL .GT. 0) THEN
         WRITE(6,5) OMEGA, ZETA, ITMAX
    5    FORMAT(//2(/70('*'))/'**',66X,'**'/'**',24X,'START SOR',
     A   ' ROUTINE:',24X,'**'/'**',66X,'**',2(/70('*'))//
     B   5X,'RELAXATION FACTOR = ',F5.2/5X,'CONVERGENCE CRITERIA = ',
     C   E10.2/5X,'MAXIMUM ITERATIONS = ',I5)
      ENDIF
C
C INITIALIZE VALUES FOR UNKN.
C
      DO 10 I = 1,MNEQ
         UNKN(I) = 0.D0
  10  CONTINUE
C
C PERFORM UP TO ITMAX SOR ITERATIONS ON SYSTEM.
C
      DO 40 ITER = 1,ITMAX
         RESMAX = 0.D0
         DO 30 I = 1,MNEQ
            A = BBBB(I)
```

```
            DO 20 J = 1,MNCO
               A = A - COEF(I,J)*UNKN(IDCO(I,J))
   20       CONTINUE
            RESID = (OMEGA/COEF(I,1))*A
            IF (RESID  .GT. RESMAX) RESMAX =  RESID
            IF (-RESID .GT. RESMAX) RESMAX = -RESID
            UNKN(I) = UNKN(I) + RESID
   30    CONTINUE
C
C DEBUGGING STATEMENTS FOR LEVEL > 2.
C
         IF (LEVEL .GT. 2) WRITE(6,31)ITER,RESMAX
   31    FORMAT(5X,'FOR ITERATION ',I5,' MAXIMUM RESIDUAL IS ',E10.4)
C
C EXIT LOOP AND REPORT # OF ITERATIONS IF CONVERGENCE CRITERIA IS MET.
C
         IF (RESMAX .LE. ZETA) THEN
            IF(LEVEL .GT. 0) WRITE(6,35)ITER
   35       FORMAT('CONVERGED IN  ',I4,'  ITERATIONS.',//)
            GO TO 50
         ENDIF
   40 CONTINUE
C
C REPORT IT CONVERGENCE CRITERIA NOT MET AFTER ITMAX ITERATIONS.
C
      IF (LEVEL .GT. 0) WRITE (6,45) ITMAX
   45 FORMAT('NO CONVERGENCE AFTER  ', I4,'  ITERATIONS.',//)
C
C IF LEVEL > 0, PRODUCE SOLUTION SUMMARY TABLES.
C
   50 IF (LEVEL .GT. 0) CALL Q8SOLN(UNKN)
      RETURN
      END
```

```
      SUBROUTINE Q8CVHD(Q,RK,TINF,H,POS,V1,V2,V3,V4,V5,DIR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C Q8CVHD PRINTS A BANNER WHEN A CONVECTIVE INTERFACE IS INVOKED.
C
      CHARACTER*8 DIR
      CHARACTER   POS,A1,A2
      A1 = 'Y'
      A2 = 'Z'
      IF (POS .EQ. 'Y' .OR. POS .EQ. 'Z') A1 = 'X'
      IF (POS .EQ. 'Z')                   A2 = 'Y'
      WRITE(6,10) DIR,POS,V1,A1,V2,V3,A2,V4,V5,Q,RK,TINF,H
   10 FORMAT(2(/73('*'))/'**',
     A 6X,A8,'  CONVECTIVE  INTERFACE  INVOKED  AT   ',A1,' = ',
     B F7.4,5X,'**'/'**',6X,A1,'  =  ',F8.4,' TO ',F8.4,'  &  ',A1,
     C '  =  ',F8.4,' TO  ',F8.4,5X,'**'/'**',6X,'Q   = ',E8.3,'  K =',
     D F7.3,'  TINF  =  ',F8.3,'  H =',F8.3,5X,'**',2(/73('*'))//)
      RETURN
      END


      SUBROUTINE Q8CDHD(QA,RKA,QB,RKB,POS,V1,V2,V3,V4,V5)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C Q8CDHD PRINTS A BANNER WHEN A CONDUCTIVE INTERFACE IS INVOKED
C
      CHARACTER   POS,A1,A2
      A1 = 'Y'
      A2 = 'Z'
      IF (POS .EQ. 'Y' .OR. POS .EQ. 'Z') A1 = 'X'
      IF (POS .EQ. 'Z')                   A2 = 'Y'
      WRITE(6,10) POS,V1,A1,V2,V3,A2,V4,V5,QA,RKA,QB,RKB
   10 FORMAT(2(/73('*'))/'**',
     A 10X,'  CONDUCTIVE  INTERFACE  INVOKED  AT   ',A1,' = ',
     B F7.4,9X,'**'/'**',6X,A1,'  =  ',F8.4,'  TO',F8.4,'  &  ',A1,
     C '  =  ',F8.4,'  TO ',F8.4,5X,'**'/'**',6X,'QA  = ',E8.3,' RKA=',
     D F7.3,'  QB    =  ',E8.3,' RKB=',F8.3,5X,'**',2(/73('*'))//)
      RETURN
      END


      SUBROUTINE Q8EQHD(NEQN)
C
C Q8EQHD IS A DEBUG SUBROUTINE THAT PRODUCES THE HEADER FOR THE EQUATION
C CURRENTLY BEING MODIFIED.
C
      WRITE (6,1) NEQN
    1 FORMAT (/24('+'),' DATA FOR EQUATION ',I4,' ',25('+'))
      RETURN
      END
```

```
      SUBROUTINE Q8EQDT(L3CVCT,NEQN,I,X,DX,A,NX,I1,IX1,IX2,J,Y,DY,B,NY,
     A       J1,JY1,JY2,K,Z,DZ,C,NZ,K1,KZ1,KZ2,COEF,IDCO,BBBB,MNEQ,MNCO)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
C Q8EQDT IS A DEBUG SUBROUTINE THAT OUTPUTS THE RELEVANT PARAMETER
C DATA USED WHEN MODIFYING THE CURRENT EQUATION.
C
      INTEGER IDCO(MNEQ,MNCO)
      DOUBLE  PRECISION  COEF(MNEQ,MNCO)
      DOUBLE  PRECISION  BBBB(MNEQ)
      DOUBLE  PRECISION  TEMP(8)
      LOGICAL L3CVCT(1:10000,6)
      CHARACTER*8 POSIT(8)
      INTEGER  EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
      DATA     EAST,SOUTH,WEST,NORTH,UPPER,LOWER,CENTER,RHS
     A       /    1,    2,    3,    4,    5,    6,    7,  8 /
      IF (L3CVCT(NEQN,EAST) .OR. L3CVCT(NEQN,WEST)) THEN
           WRITE(6,5) I, X, DX, A, NX, I1
    5      FORMAT (73('-')/'I=',I3,'  X =',F6.4,'  DX=',F6.4,
     A     '  A =',F6.4,'  NX=',I3,'  I1=',I3,'   XCVECT INTERFACE')
      ELSE
           WRITE (6,10) I, X, DX, A, NX, I1,IX1,IX2
   10      FORMAT (73('-')/'I=',I3,'  X =',F6.4,'  DX=',F6.4,
     A '  A =',F6.4,'  NX=',I3,'  I1=',I3,'   IX1=',I3,'  IX2=',I3)
      ENDIF
      IF (L3CVCT(NEQN,NORTH) .OR. L3CVCT(NEQN,SOUTH)) THEN
           WRITE (6,15) J, Y, DY, B, NY, J1
   15      FORMAT ('J=',I3,'  Y =',F6.4,'  DY=',F6.4,'  B =',
     A     F6.4,'  NY=',I3,'  J1=',I3,'   YCVECT INTERFACE')
      ELSE
           WRITE (6,20) J, Y, DY, B, NY, J1,JY1,JY2
   20      FORMAT ('J=',I3,'  Y =',F6.4,'  DY=',F6.4,'  B =',
     A     F6.4,'  NY=',I3,'  J1=',I3,'   JY1=',I3,'  JY2=',I3)
      ENDIF
      IF (L3CVCT(NEQN,UPPER) .OR. L3CVCT(NEQN,LOWER)) THEN
           WRITE (6,25) K, Z, DZ, C, NZ, K1
   25      FORMAT ('K=',I3,'  Z =',F6.4,'  DZ=',F6.4,'  C =',
     A     F6.4,'  NZ=',I3,'  K1=',I3,'   ZCVECT INTERFACE'/73('-'))
      ELSE
         IF (KZ1 .EQ. 0 .AND. KZ2 .EQ. 0) THEN
            WRITE (6,28) K, Z, DZ, C, NZ, K1
   28       FORMAT ('K=',I3,'  Z =',F6.4,'  DZ=',F6.4,'  C =',
     A      F6.4,'  NZ=',I3,'  K1=',I3,'   ZCDUCT INTERFACE'/73('-'))
         ELSE
            WRITE (6,30) K, Z, DZ, C, NZ, K1, KZ1, KZ2
   30       FORMAT ('K=',I3,'  Z =',F6.4,'  DZ=',F6.4,'  C =',
     A      F6.4,'  NZ=',I3,'  K1=',I3,'   KZ1=',I3,'  KZ2=',I3/
     B      73('-'))
         ENDIF
      ENDIF
      POSIT(1) = '  CENTER'
      POSIT(8) = '     RHS'
      M        = IDCO(NEQN,1)
      DO 35 N = 2,7
           POSIT(N) = '        '
           IF(COEF(NEQN,N) .EQ. 0.D0) GOTO 35
           IF(IDCO(NEQN,N) .EQ. M-1)     POSIT(N) = '    WEST'
           IF(IDCO(NEQN,N) .EQ. M+1)     POSIT(N) = '    EAST'
           IF(IDCO(NEQN,N) .EQ. M-NX)    POSIT(N) = '   SOUTH'
           IF(IDCO(NEQN,N) .EQ. M+NX)    POSIT(N) = '   NORTH'
```

```
            IF(IDCO(NEQN,N) .EQ. M-NX*NY) POSIT(N) = '   LOWER'
            IF(IDCO(NEQN,N) .EQ. M+NX*NY) POSIT(N) = '   UPPER'
   35 CONTINUE
      DO 40 M = 1,6
            TEMP(M)    = COEF(NEQN,M+1)/COEF(NEQN,1)
   40 CONTINUE
      TEMP(CENTER)     = COEF(NEQN,1)
      TEMP(RHS)        = BBBB(NEQN)/TEMP(CENTER)
      WRITE(6,45)(POSIT(N),N=1,8)
   45 FORMAT(3X,7A8,6X,A8)
      WRITE(6,50) TEMP(7),(TEMP(N),N=1,6),TEMP(8)
   50 FORMAT (E11.4,6F8.4,'  =  ',E9.4/73('-')/)
      RETURN
      END


      SUBROUTINE Q8BDAT(COEF, RHS, ITYPE, POS)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*5 POS
C
C THIS IS A DEBUG SUBROUTINE TO CHECK DATA GENERATED BY SUBROUTINE
C Q3BNDY.
C
      WRITE(6,10) POS, ITYPE, COEF, RHS
   10 FORMAT('--->BOUNDARY ',A5,' TYPE ',I2,'     COEF=',E12.4,
     A       '     RHS=',E12.4,'<---')
      RETURN
      END


      SUBROUTINE Q8BRST(COEF, RHS, ITYPE, POS)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*5 POS
C
C Q8BRST IS A DEBUG SUBROUTINE TO CHECK DATA GENERATED BY SUBROUTINE
C Q3BNDY WHEN A BOUNDARY IS RESET.
C
      WRITE(6,10) POS, ITYPE, COEF, RHS
   10 FORMAT('--->RST BDRY ',A5,' TYPE ',I2,'     COEF=',E12.4,
     A       '     RHS=',E12.4,'<---')
      RETURN
      END
```

```
      SUBROUTINE Q8SOLN(U)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-----------------------------------------------------------------------
C  E L L P A C K  OUTPUT  M O D U L E  SOLUTION SUMMARY TABLES
C-----------------------------------------------------------------------
C  P U R P O S E
C
C   Q8SOLN PRODUCES A SET OF SOLUTION SUMMARY TABLES THAT ARE FAIRLY EASY
C   TO READ.  A TABLE IS PRODUCED FOR EACH Z GRID POINT DEFINED FOR THE
C   REGION.  DUE TO SPACE LIMITATIONS, ONLY 9 X VARIABLES CAN BE PRINTED
C   ACROSS THE PAGE.  THE VARIABLE U IS THE SOLUTION VECTOR.
C
C   RESTRICTION -- ALL BOUNDARY CONDITIONS MUST BE NON-DIRICHLET.  IF THEY
C   ARE NOT, THE DATA TABLES WILL NOT PROPERLY FORMAT.
C
C  A U T H O R
C
C   VINCENT J. VAN JOOLEN
C
C  V E R S I O N
C
C   MAY 1991
C
C-----------------------------------------------------------------------
C
      COMMON / C1IVDI / NEQN, MNEQ, NCOE, MNCO
      COMMON / C1IVGR / NX, NY, NZ, I1NBPT, I1MBPT, I1PACK
      COMMON / C1GRDX / GRDX(1)
      COMMON / C1GRDY / GRDY(1)
      COMMON / C1GRDZ / GRDZ(1)
      DOUBLE PRECISION U(MNEQ)
      INTEGER N(9), EQN(9)
C
C INITIALIZE N VALUES TO BE USED IN WRITE STATEMENTS.
C
      DO 5 I = 1,9
         N(I) = I*NX/9
    5 CONTINUE
C
C PRINT SOLUTION TABLE HEADER.
C
      WRITE(6,10)
   10 FORMAT(2(/70('*'))/'**',66X,'**'/'**',15X,'SOLUTION TABLE',
     A' FOR ELLIPTIC EQUATION',15X,'**'/'**',66X,'**',2(/70('*')))
C
C GENERATE NZ TABLES FOR EACH Z GRID POINT.
C
      DO 50 I = 1,NZ
C
C WRITE HEADER FOR EACH VALUE OF Z.
C
         WRITE(6,15) GRDZ(I)
   15    FORMAT(///14('>'),5X,'TEMPERATURE PROFILE AT Z = ',F7.4,5X,
     A   14('<')//)
         WRITE(6,20)(GRDX(N(J)),J=1,9)
   20    FORMAT(4X,'X =  ',9F7.4/'Y =')
         DO 45 J = 1,NY
            DO 25 K = 1,9
               EQN(K) = N(K) + NX*((J-1) + NY*(I-1))
```

```
25          CONTINUE
            WRITE(6,30)GRDY(J),(U(EQN(K)), K=1,9)
30          FORMAT(F7.4,2X,9(F7.2))
45       CONTINUE
50    CONTINUE
      WRITE(6,60)
60    FORMAT(///)
      RETURN
      END

END.
```

APPENDIX E

LEVEL 1 OUTPUT FOR A SINGLE CHIP ON A SUBSTRATE

```
------------------------
 DISCRETIZATION MODULE
------------------------

   7 P O I N T   3 D

   NUMBER OF EQUATIONS          648

   EXECUTION SUCCESSFUL

**********************************************************************************
**********************************************************************************
**      UPPER     CONVECTIVE  INTERFACE  INVOKED  AT   Z =  0.0005      **
**      X  =    0.0000 TO   0.0250  &  Y  =    0.0000 TO    0.0250      **
**      Q   = .000E+00  K =  1.000  TINF  =    20.000  H =  50.000      **
**********************************************************************************
**********************************************************************************


**********************************************************************************
**********************************************************************************
**      EASTERN   CONVECTIVE  INTERFACE  INVOKED  AT   X =  0.0010      **
**      Y  =    0.0000 TO   0.0010  &  Z  =    0.0005 TO    0.0025      **
**      Q   = .625E+08  K = 50.000  TINF  =    20.000  H =  50.000      **
**********************************************************************************
**********************************************************************************


**********************************************************************************
**********************************************************************************
**      NORTHERN  CONVECTIVE  INTERFACE  INVOKED  AT   Y =  0.0010      **
**      X  =    0.0000 TO   0.0010  &  Z  =    0.0005 TO    0.0025      **
**      Q   = .625E+08  K = 50.000  TINF  =    20.000  H =  50.000      **
**********************************************************************************
**********************************************************************************


**********************************************************************************
**********************************************************************************
**           CONDUCTIVE  INTERFACE  INVOKED  AT   Z =  0.0005           **
**      X  =    0.0000  TO  0.0010  &  Y  =    0.0000  TO   0.0010      **
**      QA  = .000E+00 RKA=  1.000  QB    =  .625E+08 RKB=  50.000      **
**********************************************************************************
**********************************************************************************
```

```
**********************************************************************
**********************************************************************
**                                                                  **
**                        START SOR ROUTINE:                        **
**                                                                  **
**********************************************************************
**********************************************************************

     RELAXATION FACTOR     =   1.94
     CONVERGENCE CRITERIA  =   0.10E-03
     MAXIMUM ITERATIONS    = 10000

     CONVERGED IN   227  ITERATIONS.


**********************************************************************
**********************************************************************
**                                                                  **
**              SOLUTION TABLE FOR ELLIPTIC EQUATION                **
**                                                                  **
**********************************************************************
**********************************************************************
```

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0000 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	68.63	68.46	67.83	65.52	41.93	22.90	20.26	20.02	20.00
0.0003	68.46	68.30	67.66	65.35	41.76	22.89	20.26	20.02	20.00
0.0006	67.83	67.66	67.02	64.73	41.23	22.85	20.26	20.02	20.00
0.0010	65.52	65.35	64.73	62.53	39.95	22.76	20.26	20.02	20.00
0.0020	41.93	41.76	41.23	39.95	33.73	22.39	20.24	20.02	20.00
0.0060	22.90	22.89	22.85	22.76	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0001 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	68.92	68.75	68.14	65.89	42.02	22.91	20.26	20.02	20.00
0.0003	68.75	68.59	67.97	65.73	41.85	22.90	20.26	20.02	20.00
0.0006	68.14	67.97	67.36	65.13	41.32	22.86	20.26	20.02	20.00
0.0010	65.89	65.73	65.13	62.99	40.03	22.78	20.26	20.02	20.00
0.0020	42.02	41.85	41.32	40.03	33.78	22.39	20.24	20.02	20.00
0.0060	22.91	22.90	22.86	22.78	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0003 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	69.70	69.59	69.17	67.53	42.09	22.92	20.26	20.02	20.00
0.0003	69.59	69.48	69.06	67.42	41.92	22.91	20.26	20.02	20.00
0.0006	69.17	69.06	68.64	67.01	41.39	22.87	20.26	20.02	20.00
0.0010	67.53	67.42	67.01	65.41	40.11	22.78	20.26	20.02	20.00
0.0020	42.09	41.92	41.39	40.11	33.81	22.40	20.24	20.02	20.00
0.0060	22.92	22.91	22.87	22.78	22.40	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0005 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	70.67	70.65	70.61	70.48	41.99	22.90	20.26	20.02	20.00
0.0003	70.65	70.64	70.59	70.46	41.82	22.89	20.26	20.02	20.00
0.0006	70.61	70.59	70.55	70.41	41.29	22.85	20.26	20.02	20.00
0.0010	70.48	70.46	70.41	70.26	40.03	22.76	20.26	20.02	20.00
0.0020	41.99	41.82	41.29	40.03	33.73	22.39	20.24	20.02	20.00
0.0060	22.90	22.89	22.85	22.76	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0007 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	70.66	70.65	70.61	70.54	20.00	20.00	20.00	20.00	20.00
0.0003	70.65	70.64	70.60	70.53	20.00	20.00	20.00	20.00	20.00
0.0006	70.61	70.60	70.56	70.48	20.00	20.00	20.00	20.00	20.00
0.0010	70.54	70.53	70.48	70.40	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0011 <<<<<<<<<<<<<<

X =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
Y =									
0.0000	70.57	70.56	70.53	70.49	20.00	20.00	20.00	20.00	20.00
0.0003	70.56	70.55	70.52	70.47	20.00	20.00	20.00	20.00	20.00
0.0006	70.53	70.52	70.48	70.42	20.00	20.00	20.00	20.00	20.00
0.0010	70.49	70.47	70.42	70.35	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0018 <<<<<<<<<<<<<<

X =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
Y =									
0.0000	70.14	70.11	70.04	69.96	20.00	20.00	20.00	20.00	20.00
0.0003	70.11	70.08	70.00	69.89	20.00	20.00	20.00	20.00	20.00
0.0006	70.04	70.00	69.87	69.69	20.00	20.00	20.00	20.00	20.00
0.0010	69.96	69.89	69.69	69.30	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0025 <<<<<<<<<<<<<<

X =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
Y =									
0.0000	69.68	69.61	69.44	69.21	20.00	20.00	20.00	20.00	20.00
0.0003	69.61	69.52	69.28	68.91	20.00	20.00	20.00	20.00	20.00
0.0006	69.44	69.28	68.79	67.74	20.00	20.00	20.00	20.00	20.00
0.0010	69.21	68.91	67.74	63.12	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

```
-----------------
 SOLUTION MODULE
-----------------

   B A N D   G E

   NUMBER OF EQUATIONS       648
   LOWER BANDWIDTH            81
   UPPER BANDWIDTH            81
   REQUIRED WORKSPACE     158760

   EXECUTION SUCCESSFUL

**************************************************************************
**************************************************************************
**                                                                      **
**               SOLUTION TABLE FOR ELLIPTIC EQUATION                   **
**                                                                      **
**************************************************************************
**************************************************************************
```

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0000 <<<<<<<<<<<<<<

Y = \ X =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	68.64	68.47	67.83	65.52	41.93	22.90	20.26	20.02	20.00
0.0003	68.47	68.30	67.66	65.35	41.76	22.89	20.26	20.02	20.00
0.0006	67.83	67.66	67.02	64.74	41.23	22.85	20.26	20.02	20.00
0.0010	65.52	65.35	64.74	62.53	39.95	22.76	20.26	20.02	20.00
0.0020	41.93	41.76	41.23	39.95	33.73	22.39	20.24	20.02	20.00
0.0060	22.90	22.89	22.85	22.76	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0001 <<<<<<<<<<<<<<

Y = \ X =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	68.92	68.75	68.14	65.89	42.02	22.91	20.26	20.02	20.00
0.0003	68.75	68.59	67.97	65.73	41.85	22.90	20.26	20.02	20.00
0.0006	68.14	67.97	67.36	65.14	41.32	22.86	20.26	20.02	20.00
0.0010	65.89	65.73	65.14	62.99	40.04	22.78	20.26	20.02	20.00
0.0020	42.02	41.85	41.32	40.04	33.78	22.39	20.24	20.02	20.00
0.0060	22.91	22.90	22.86	22.78	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0003 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	69.70	69.59	69.17	67.53	42.09	22.92	20.26	20.02	20.00
0.0003	69.59	69.48	69.06	67.42	41.92	22.91	20.26	20.02	20.00
0.0006	69.17	69.06	68.65	67.01	41.39	22.87	20.26	20.02	20.00
0.0010	67.53	67.42	67.01	65.41	40.11	22.78	20.26	20.02	20.00
0.0020	42.09	41.92	41.39	40.11	33.81	22.40	20.24	20.02	20.00
0.0060	22.92	22.91	22.87	22.78	22.40	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0005 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	70.67	70.65	70.61	70.48	41.99	22.90	20.26	20.02	20.00
0.0003	70.65	70.64	70.59	70.46	41.82	22.89	20.26	20.02	20.00
0.0006	70.61	70.59	70.55	70.41	41.29	22.85	20.26	20.02	20.00
0.0010	70.48	70.46	70.41	70.26	40.03	22.76	20.26	20.02	20.00
0.0020	41.99	41.82	41.29	40.03	33.73	22.39	20.24	20.02	20.00
0.0060	22.90	22.89	22.85	22.76	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0007 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	70.66	70.65	70.62	70.54	20.00	20.00	20.00	20.00	20.00
0.0003	70.65	70.64	70.60	70.53	20.00	20.00	20.00	20.00	20.00
0.0006	70.62	70.60	70.56	70.49	20.00	20.00	20.00	20.00	20.00
0.0010	70.54	70.53	70.49	70.40	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0011 <<<<<<<<<<<<<<

X =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
Y =									
0.0000	70.57	70.56	70.53	70.49	20.00	20.00	20.00	20.00	20.00
0.0003	70.56	70.55	70.52	70.47	20.00	20.00	20.00	20.00	20.00
0.0006	70.53	70.52	70.48	70.42	20.00	20.00	20.00	20.00	20.00
0.0010	70.49	70.47	70.42	70.35	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0018 <<<<<<<<<<<<<<

X =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
Y =									
0.0000	70.14	70.11	70.05	69.96	20.00	20.00	20.00	20.00	20.00
0.0003	70.11	70.08	70.00	69.89	20.00	20.00	20.00	20.00	20.00
0.0006	70.05	70.00	69.87	69.69	20.00	20.00	20.00	20.00	20.00
0.0010	69.96	69.89	69.69	69.30	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0025 <<<<<<<<<<<<<<

X =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
Y =									
0.0000	69.68	69.61	69.44	69.21	20.00	20.00	20.00	20.00	20.00
0.0003	69.61	69.52	69.28	68.91	20.00	20.00	20.00	20.00	20.00
0.0006	69.44	69.28	68.79	67.74	20.00	20.00	20.00	20.00	20.00
0.0010	69.21	68.91	67.74	63.12	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

```
----------------
 ELLPACK OUTPUT
----------------
```

```
++++++++++++++++++++++++++
+                        +
+    EXECUTION  TIMES    +
+                        +
++++++++++++++++++++++++++
```

SECONDS	MODULE NAME
0.07	7-POINT 3D
0.07	BAND GE SETUP
1.17	BAND GE
2.01	TOTAL TIME

APPENDIX F

SAMPLE LEVEL 2 OUTPUT

LEVEL 2 output is extensive and should be used exclusively for debugging. The program written for this paper provides LEVEL 2 output at the time the equations are modified along the interfaces and during each iteration of the SOR algorithm. A sample of this output is listed in this appendix.

For an equation lying on an interface which is being modified, a typical LEVEL 2 data set might look like:

```
++++++++++++++++++++++++++ DATA FOR EQUATION  244 ++++++++++++++++++++++++++
--->BOUNDARY  WEST TYPE  2     COEF=  0.5556E+07     RHS=  0.0000E+00<---
--->BOUNDARY SOUTH TYPE  2     COEF=  0.5556E+07     RHS=  0.0000E+00<---
------------------------------------------------------------------------------
I=  1  X =0.0000  DX=0.0003  A =1.0000  NX=  9  I1=  1   IX1=  1  IX2=  9
J=  1  Y =0.0000  DY=0.0003  B =1.0000  NY=  9  J1=  1   JY1=  1  JY2=  9
K=  4  Z =0.0005  DZ=0.0002  C =1.0000  NZ=  8  K1=  1   ZCVECT INTERFACE
------------------------------------------------------------------------------
     CENTER     EAST    NORTH    UPPER                                    RHS
-0.4747E+08 -0.2341 -0.2341 -0.5266  0.0000  0.0000  0.0000  =  .1053E+00
------------------------------------------------------------------------------
```

The header indicates that the coefficients are being modified for the equation whose index number is 244.

The first section of the data pertains to the boundary. The particular grid point under investigation lies on a type 2 (Neumann) western and southern boundary. The western and southern coefficients are listed as they appear before they are normalized by

the central variable coefficient. The right hand side of the boundary, in both cases 0, is also listed.

The second section of the data reports the parameters used to modify the equation. They are defined as follows:

- I: Index of the x grid point for the equation listed.
- J: Index of the y grid point for the equation listed.
- K: Index of the z grid point for the equation listed.
- X: X grid point.
- Y: Y grid point.
- Z: Z grid point.
- DX: Distance between current x point and previous x point on the grid. Equals 1 if on western boundary.
- DY: Distance between current y point and previous y point on the grid. Equals 1 if on southern boundary.
- DZ: Distance between current z point and previous z point on the grid. Equals if on lower boundary.
- A: Alpha multiplier for east x interval.
- B: Beta multiplier for north y interval.
- C: Gamma multiplier for upper z interval.
- NX: Total number of x grid points.
- NY: Total number of y grid points.
- NZ: Total number of z grid points.

- I1: Position of first x grid point where the temperature is unknown (Equals 2 if boundary is Dirichlet).
- J1: Position of first y grid point where the temperature is unknown (Equals 2 if boundary is Dirichlet).
- K1: Position of first z grid point where the temperature is unknown (Equals 2 if boundary is Dirichlet).
- IX1: Index of first x grid point on the interface.
- IX2: Index of last x grid point on the interface.
- JY1: Index of first y grid point on the interface.
- JY2: Index of last y grid point on the interface.

The line 'ZCVECT INTERFACE' indicates that the point lies on a convective interface parallel to the xy plane.

The final section contains the modified values of the coefficients of the equation. The number marked 'center' is the value of the center coefficient after it has passed through the interface modification routines. The remaining numbers are the values of the respective coefficients after they have been normalized by dividing by the center coefficient. The right hand side values are also printed after being normalized to the center coefficient.

During the execution of the SOR subroutine, a typical LEVEL 2 output appears as:

```
FOR ITERATION      1 MAXIMUM RESIDUAL IS 0.3889E+02
 .      .        .     .       .      .      .
 .      .        .     .       .      .      .
FOR ITERATION    227 MAXIMUM RESIDUAL IS 0.9671E-04
```

This output reports the maximum residual for each iteration and allows the user to track the convergence of the program.

In both examples, the LEVEL 2 output proved extremely useful when debugging the program. It should be noted that the LEVEL 1 output will also appear when LEVEL 2 is requested.

APPENDIX G

LEVEL 0 OUTPUT FOR THE INFINITE SUBSTRATE/CHIP MODEL

```
**************************************************************************
**************************************************************************
**                                                                      **
**                SOLUTION TABLE FOR ELLIPTIC EQUATION                  **
**                                                                      **
**************************************************************************
**************************************************************************
```

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0000 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	68.64	68.47	67.83	65.52	41.93	22.90	20.26	20.02	20.00
0.0003	68.47	68.30	67.66	65.35	41.76	22.89	20.26	20.02	20.00
0.0006	67.83	67.66	67.02	64.74	41.23	22.85	20.26	20.02	20.00
0.0010	65.52	65.35	64.74	62.53	39.95	22.76	20.26	20.02	20.00
0.0020	41.93	41.76	41.23	39.95	33.73	22.39	20.24	20.02	20.00
0.0060	22.90	22.89	22.85	22.76	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0001 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	68.92	68.75	68.14	65.89	42.02	22.91	20.26	20.02	20.00
0.0003	68.75	68.59	67.97	65.73	41.85	22.90	20.26	20.02	20.00
0.0006	68.14	67.97	67.36	65.14	41.32	22.86	20.26	20.02	20.00
0.0010	65.89	65.73	65.14	62.99	40.04	22.78	20.26	20.02	20.00
0.0020	42.02	41.85	41.32	40.04	33.78	22.39	20.24	20.02	20.00
0.0060	22.91	22.90	22.86	22.78	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0003 <<<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	69.70	69.59	69.17	67.53	42.09	22.92	20.26	20.02	20.00
0.0003	69.59	69.48	69.06	67.42	41.92	22.91	20.26	20.02	20.00
0.0006	69.17	69.06	68.65	67.01	41.39	22.87	20.26	20.02	20.00
0.0010	67.53	67.42	67.01	65.41	40.11	22.78	20.26	20.02	20.00
0.0020	42.09	41.92	41.39	40.11	33.81	22.40	20.24	20.02	20.00
0.0060	22.92	22.91	22.87	22.78	22.40	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0005 <<<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	70.67	70.65	70.61	70.48	41.99	22.90	20.26	20.02	20.00
0.0003	70.65	70.64	70.59	70.46	41.82	22.89	20.26	20.02	20.00
0.0006	70.61	70.59	70.55	70.41	41.29	22.85	20.26	20.02	20.00
0.0010	70.48	70.46	70.41	70.26	40.03	22.76	20.26	20.02	20.00
0.0020	41.99	41.82	41.29	40.03	33.73	22.39	20.24	20.02	20.00
0.0060	22.90	22.89	22.85	22.76	22.39	20.67	20.09	20.01	20.00
0.0120	20.26	20.26	20.26	20.26	20.24	20.09	20.02	20.00	20.00
0.0180	20.02	20.02	20.02	20.02	20.02	20.01	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0007 <<<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	70.66	70.65	70.62	70.54	20.00	20.00	20.00	20.00	20.00
0.0003	70.65	70.64	70.60	70.53	20.00	20.00	20.00	20.00	20.00
0.0006	70.62	70.60	70.56	70.49	20.00	20.00	20.00	20.00	20.00
0.0010	70.54	70.53	70.49	70.40	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0011 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	70.57	70.56	70.53	70.49	20.00	20.00	20.00	20.00	20.00
0.0003	70.56	70.55	70.52	70.47	20.00	20.00	20.00	20.00	20.00
0.0006	70.53	70.52	70.48	70.42	20.00	20.00	20.00	20.00	20.00
0.0010	70.49	70.47	70.42	70.35	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0018 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	70.14	70.11	70.05	69.96	20.00	20.00	20.00	20.00	20.00
0.0003	70.11	70.08	70.00	69.89	20.00	20.00	20.00	20.00	20.00
0.0006	70.05	70.00	69.87	69.69	20.00	20.00	20.00	20.00	20.00
0.0010	69.96	69.89	69.69	69.30	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

>>>>>>>>>>>>>> TEMPERATURE PROFILE AT Z = 0.0025 <<<<<<<<<<<<<<

X = / Y =	0.0000	0.0003	0.0006	0.0010	0.0020	0.0060	0.0120	0.0180	0.0250
0.0000	69.68	69.61	69.44	69.21	20.00	20.00	20.00	20.00	20.00
0.0003	69.61	69.52	69.28	68.91	20.00	20.00	20.00	20.00	20.00
0.0006	69.44	69.28	68.79	67.74	20.00	20.00	20.00	20.00	20.00
0.0010	69.21	68.91	67.74	63.12	20.00	20.00	20.00	20.00	20.00
0.0020	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0060	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0120	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0180	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
0.0250	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

LIST OF REFERENCES

1. Haberman, R., *Elementary Applied Partial Differential Equations*, Prentice Hall, Inc., 1987.

2. Özişik, M. N., *Basic Heat Transfer*, McGraw Hill Book Company, 1977.

3. *386-MATLAB™*, The MathWorks, Inc., 1990.

4. Rice, J. R., and Boisvert, R. F., *Solving Elliptic Problems Using ELLPACK*, Springer Verlag, 1984.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3. Professor H. M. Fredricksen, Code MA/Fs Department of Mathematics Naval Postgraduate School Monterey, CA 93943	1
4. Professor B. Neta, Code MA/Nd Department of Mathematics Naval Postgraduate School Monterey, CA 93943	4
5. Professor R. H. Franke, Code MA/Fe Department of Mathematics Naval Postgraduate School Monterey, CA 93943	1
6. Professor Y. Joshi, Code ME/Ji Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	1

7. Commanding Officer
Naval Weapons Support Center
Bldg. 2044, Code 6042
Crane, IN 47522-5060
ATTN: Mr. A. Buechler
1

8. Professor J.R. Rice
Department of Computer Science
Purdue University
W. Lafayette, IN 47907
1

9. Professor Elias N. Houstis
Department of Computer Science
Purdue University
W. Lafayette, IN 47907
1

10. LT Vincent J. van Joolen
Department Head Class 120
SWOSCOLCOM
Newport, RI 02841-5012
1

END